DOCUMENT RESUME

ED 304 325 SE 050 384

TITLE East Coast Logo Conference '87 Proceedings (Arlington, Virginia, April 2-4, 1987).
INSTITUTION International Council for Computers in Education, Eugene, Oreg.
REPORT NO ISBN-0-88736-195-1
PUB DATE 87
NOTE 247p.; Small print and drawings may not reproduce well.
PUB TYPE Collected Works - Conference Proceedings (021)

EDRS PRICE MF01/PC10 Plus Postage.
DESCRIPTORS *Computer Assisted Instruction; Computer Graphics; Computers; *Computer Software; *Computer Uses in Education; Elementary Secondary Education; *Instructional Materials; Mathematical Enrichment; Mathematics Materials; *Science Activities; Science Materials
IDENTIFIERS *LOGO Programing Language; LOGO System

ABSTRACT

A total of 59 papers are compiled into these proceedings. The papers are organized alphabetically by each author's last name. A directory of speakers' names and addresses is included. In the index of this publication, papers are listed by session. General sessions are listed first, followed by 21 regular sessions: (1) "Logo and Music"; (2) "Logo Interactions"; (3) "Logo: FD to the Future"; (4) "Logo Connections"; (5) "Logo Environments"; (6) "Secondary Logo Integration"; (7) "Logolike Software and Other Compelling Media"; (8) "Special Goldenberg Session"; (9) "Building with Logo"; (10) "Logo and Robotics"; (11) "Logo in the Early Years"; (12) "Logo International"; (13) "Special Session: Vertical Logo Integration"; (14) "Elementary and Middle School Curriculum Applications for Logo"; (15) "Communicating through a Keyboard"; (16) "Thresholds and Ceilings"; (17) "Student Presentations"; (18) "Microcomputer Based Laboratories"; (19) "From the Classroom"; (20) "The Big Picture"; and (21) "Student Activities." There are nine "paper sessions" included. (YP)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED304325

EAST COAST LOGO CONFERENCE '87

April 2-4, 1987
Arlington, Virginia

CONFERENCE PROCEEDINGS

Tom Lough and Glen Bull
Co-Chairmen

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☒ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

SE 050 384

Meckler Publishing Corporation

2

ECLC '87

EAST COAST LOGO CONFERENCE '87

April 2-4, 1987
Arlington, Virginia

CONFERENCE PROCEEDINGS

Tom Lough and Glen Bull
Co-Chairmen

Meckler Publishing Corporation

Copyright © 1987 by Meckler Publishing Corporation

All rights reserved. No part of this publication may be reproduced in any form by any means whatsoever without prior permission in writing from the publisher.

ISBN 0-88736-195-1

MECKLER PUBLISHING CORPORATION, 11 Ferry Lane West, Westport, CT 06880
MECKLER PUBLISHING, 3 Henrietta Street, London WC2E 8LU, UK

Printed and bound in the United States of America

CURRY SCHOOL OF EDUCATION
DEPARTMENT OF EDUCATIONAL STUDIES

April 2, 1987

Dear ECLC '87 Attendee:

We have asked the speakers at the East Coast Logo Conference '87 to provide three to five pages of material which contain information about their presentations. The papers have been compiled into these proceedings of more than two hundred pages.

We invite you to use these proceedings to help you select which presentations to attend during the conference. However, we anticipate that the proceedings will become a valuable Logo reference after the conference.

The papers in the proceedings are organized alphabetically by each author's last name. A table of contents in the front contains a similar alphabetical listing. An index in the back lists the papers in session order, providing a close correlation with the conference program. Should you desire to contact any speaker by mail, a directory of their names and addresses is included.

We would like to thank all of the speakers at ECLC '87 for their generous contributions of their time and experiences. We especially appreciate their willingness to provide these papers in advance of the conference.

FD 100!

Tom Lough
ECLC '87
Co-Chair

Glen Bull
ECLC '87
Co-Chair

EDUCATIONAL EVALUATION EDUCATIONAL PSYCHOLOGY EDUCATIONAL RESEARCH
INSTRUCTIONAL TECHNOLOGY SOCIAL FOUNDATIONS OF EDUCATION
RUFFNER HALL, UNIVERSITY OF VIRGINIA, 405 EMMET STREET, CHARLOTTESVILLE, VA. 22903-2495

Contents

Adamson, Eadie
LogoWriter: A Learning Environment and a Programming Language for All Levels 1

Addicks, Tom
Logo and Computer Assisted Design (CAD) 3

Adelberg, Steven M.
Creative Use of Logo with Special Student Populations: Using Atari Logo in the Primary Grades at Bailey's Elementary School with ESL Students 8

Anderson, Scott (Presented by Roseanne Kramer)
Fantavision in Education 11

Bailey, Suzanne L.
Logo Activities and Strategies in the Kindergarten 13

Ballew, Hunter
Geometry through LOGO for Teachers in the Middle Grades 15

Batson, Trent
The ENFI Project 19

Bauserman, Jane, and Dianne Lawrence
One Computer—Thirty Kids—And You Want Me to Use Logo?? 23

Birch, Alison
Creating a Learning Environment for Teachers 27

Birch, Alison
Playing with Language through Logo Projects 31

Bull, Glen L., and Paula S. Cochran
A Book with 50,000 Pictures: Logo and Videodiscs 34

Calvert, Betsy [See Wissick, Cheryl, and Betsy Calvert]

Carter, Ricky
The Logo Algebra Project 39

Cartmell, David J.
Talking with IBM Logo 43

Chesebrough, David E.
Logo Integration into an Entire School System 47

Cochran, Paula S. [See Bull, Glen L., and Paula S. Cochran]

Collison, George [See D'Introno, Lis, et al.]

Cory, Sheila
The Mathematical Turtle: Logo in the 4–8 Math Curriculum 54

Crowley, John E. (Presented by Catherine Helgoe)
No Limits LEGO TC logo..... 58

Cunningham, Beverly and Lee
Associated Learning: Logo Primitives in Real Life 61

Cunningham, Beverly and Lee
Ten Tried and True Turtle Technical Tips 65

D'Introno, Lis, Stephen D. Mosher, Christopher Van Leeuwen, and George Collison
A Logo Curriculum Based on Critical Thinking and Problem Solving Using the Polya Model 68

Evans, Andrew [See Glass, Jonell, and Andrew Evans]

Fairless, Mary [See Nye, Barbara]

Fitch, Dorothy M.
Logo Learning Tools for the Classroom 72

Foster, America H.
Creating a LOGO Environment in a Geometry Class 75

Foster, Jacqueline M.
The Platonic Solids 80

Freyd, Pamela
Reading, Writing, and Logo: The Ultimate Language Experience 84

Fry, James S.
Logo Music Tools: "MusicLand" 87

Gelman, Michael A.
Methods of Simulating a Robot's Behavior Using Logo 91

Glass, Jonell, and Andrew Evans
Logo Quickly 96

Goldenberg, E. Paul
Language Science, Language Play: Using Logo to Explore What You Know 99

Goldenberg, E. Paul
Making Functions Concrete: Explorations for Algebra and Pre-algebra 104

Hain, Stephen [See Krakowsky, Phillipe, and Stephen Hain]

Hanson, Janet Bard
The Turtle Grows Up: LOGO Applications in High School Physics, English, Biology, and Artificial Intelligence 111

Harris, Judi
Logo LinX with Traditional K–8 Curricula 115

Harvey, Francis A.
Tool Procedures for Efficient Development of Logo-Based Applications: Text Screen Formatting Tools 119

Helgoe, Catherine [See Crowley, John]

Heller, Rachelle [Martin, C. Dianne, and Rachelle Heller]

Kramer, Rosanne [See Anderson, Scott]

Lang, Patricia [See Olson, Judith, and Patricia Lang]

Lawrence, Dianne [See Bauserman, Jane, and Dianne Lawrence]

Lawrence, Gayle
Logo: Where Models and Reality Connect 123

Martin, C. Dianne, and Rachelle S.Heller
Educational Computing in the Soviet Union 128

McPeak, Sean, Kevin, and Jon Campbell
Student Designed Tutorials 133

Melnick, Harold R.
"Tools for Making Meaning out of Mathematics"—A Rationale 136

Mosher, Stephen D. [See D'Introno, Lis, et al.]

Muir, Robs John
Apropos Logo and Telecommunications 141

Muir, Robs John
No Threshold? No Ceiling? 144

Muir, Robs John
Secondary Logo Opportunities 146

Novak, Dan
Teaching Methods of Interfacing Logo Procedures 150

Nye, Barbara A., with Andre Rossi (Presented by Barbara West, Mary Fairless, and Andre Rossi)
Let the Valiant Turtle Robot Lead the Way to Logo! 154

Olson, Judith, and Patricia Lang
Cross Over the Bridge—A Journey to Abstraction with Logo and Manipulatives 158

9

Poplin, Rebecca L.
An Environment for Action 163

Rawitsch, Peter
BUDGET-MINDEDSTORMS 166

Ring, Curtis
Adventuring in Logo 168

Rosegrant, Teresa
Exploring the Algorithm of Written English 171

Rossi, André [See Nye, Barbara]

Silverman, Brian
Logo in the Microworld of Life 172

Silverman, Brian, and Michael Tempel
Logo Thoughts: LogoWriter and Logo II 173

Stager, Gary S.
If There Had Been Logo. . .I Might Have Learned Math 175

Tempel, Michael [See Silverman, Brian, and Michael Tempel]

Templar, Chris
Integrating Logo into the Elementary School Curriculum 180

Thyberg, Leslie F.
The Basis for a Process Approach to Programming in Logo 183

Thyberg, Leslie F.
Young Users of Logo—The Challenge and the Opportunity 187

Tipps, Steve
Logo Beyond the Square 191

Upitis, Rena B.
Logo-Like Music Environments 195

Van Leeuwen, Christopher [See D'Introno, Lis, et al.]

Vining, Madge Alexander
SeeLogo: A New Perspective on an Old Friend 199

Watt, Molly, and Daniel H. Watt
The Logo and Computers in Education Workshop: A Report on a Teacher Training Project at The People's Education Press, Beijing, China 203

Watt, Daniel H. and Molly Watt
Step by Step: East Coast Logo Conference '87 Kickoff Session 207

West, Barbara [See Nye, Barbara]

White, James E.
Symbolic Mathematics and LOGO 212

Wissick, Cheryl A.
Kids to Kids: USA and USSR 218

Wissick, Cheryl A., and Betsy Calvert
Turtle Steps to Single Switch Computer Access 222

Speakers 227

Index of Papers by Session 231

LogoWriter: A Learning Environment and a Programming Language for All Levels

Eadie Adamson
Computer Coordinator
The Allen-Stevenson School
132 East 78th Street
New York, NY 10021

An ideal learning environment stimulates creativity, thinking, and planning. A Logo environment should do all this, but most Logo environments don't do this without a lot of teacher help, largely because of the technical things which the user must understand before being able to work independently. What makes LogoWriter different from other versions of Logo is that these technical problems have, for the most part, been done away with.

In using Logo at the elementary level, one needs to consider the user population and evaluate the different versions of Logo accordingly. Therefore, one needs to take into account the thought processes and associations made by children -- and even young or not-so-young adults -- as they have beginning and continuing encounters with a computer. We all tend to forget the frightening experience of our own first encounters with a computer or to brush aside those awful memories of being left alone to deal with a program we simply couldn't figure out. Yet when we teach young children to use computers these experiences should be foremost in our minds. What can relieve this strange feeling of alienation, of something out of control? How can we make this a comfortable experience, something which these new users can relate to?

What is more familiar, even to the youngest individual, than a book? This is the metaphor upon which LogoWriter is based. This is what makes using LogoWriter so comfortable: most of the tasks involved in using it can be related to something already familiar to the user. A very important part of learning is the ability to associate a new idea with an old one. Starting something new with LogoWriter involves choosing a new page. Moving to the editor means "flipping" or turning the page. This is an acceptable thought process easily visualized by a young child. On the other hand, moving to the editor in other versions of Logo by typing ED is not easily understood, nor is loading a program. Where are you going and what's happening (forget about why)? With LogoWriter, if you flip the page, you move to the "other side" and what can be clearer than that? To go back, you flip back. To choose a new page you press return (or type NEWPAGE on the page you're using) and voila! a new page appears.

The ability to select and then cut or copy text, whether it is writing or lines of programming, represents an enormous forward step even for the most experienced programmer. How many times have you tried things out without writing everything in procedures? How successful have you been in keeping accurate notes, not to mention how successful and accurate your students, whatever their age, have been in attempting the same feat? How often have you seen a child create something wonderful and had no hope, without a complicated process, of even saving the drawing itself, not to mention the process involved in its creation? How wonderful to have a program that makes saving a drawing automatic and that also allows you to grab from the screen those wonderful things programmed in the immediate mode so that they may indeed become procedures infinitely reproducible to a child's pride and delight? What a wonderful all-purpose response to the question, "How do you get out of this?" -- ESCAPE!

At what level might a Logo version such as this, leaving aside for the moment the word-processing capability, be used? It would seem to me that, while some of us may bewail missing primitives (I'm among them) and not wish to respond happily to the challenge "but you can write that in LogoWriter," that even a "high-level" course in Logo might progress quite far before finding anything missing. There will be a little frustration with disk space, but learning to use a scrapbook disk and to organize files would fit well into an advanced course. And along the way, this high level course might find that the prospect of dealing with multiple objects more than compensated for what might be found lacking. Problems with space? Creative use of CT and CC within procedures could conceivably go a long way to alleviate some of those problems, as could the ability to access another page when necessary. That would represent a wonderful programming challenge when it arose. And what about the creative programming possibilities of creating a new copy of something at a point, then erasing what's no longer necessary and continuing a longer program? It seems to me there are plenty of challenges and a long way to go before the ceiling is reached!

What would I like to see in the best of all possible versions of Logo? A lot that is already contained within LogoWriter. I would also like the ability to program more complex shapes (both numbers of shapes and degree of complexity per shape), I'd like (of course) more memory per file, I'd like more sophisticated word-processing (how about pagination, ability to show pages on the screen, more printing options, dare I ask for footnotes?). I'd like more turtles. I'd like to have the missing primitives restored (are there really that many missing?) rather than be challenged to write them myself (or figure out a way around them, one of my favorite tasks). I see using multiple turtles as remarkably akin to some of the principles of Object Logo. I'd want that too, either as an integral part of my ideal Logo, or as a special tool file I could call up at will. I'd like more colors on the Apple. I'd like -- sometimes -- to be able to use my old Turtle Tot, even if it is a relic, and control it from the keyboard with LogoWriter. I'd like more voices to program for sound. Could I have speech also, please, as long as I'm dreaming? Some people are worried about telecommunications. I'd like that too.

Logo and Computer Assisted Design (CAD)

Tom Addicks

Berkshire Country Day School, Lenox, MA.

Beginning users of Logo are often taught the graphics commands and students begin to explore the nature of the turtle and what Logo will do. The REPEAT command instantly brings to the screen repeated patterns. Logo users modify the patterns into designs while learning about the "total turtle trip". Children are delighted with the power that Logo give them in regard to the possible patterns, colors and the like. The excitement is often dashed when students are urged to make a realistic picture. Although the task of depicting something real is often easier than developing designs, the fact that real pictures have to conform to reality limits spontaneity. Using Logo to simulate Computer Assisted Design (CAD) can put the spontaneity back into turtle graphics which depicts the real world.

As the Logo student takes the role of the architect, he must develop an idea for his drawing. The idea is limited by two factors: 1) The student's knowledge of Logo, and 2) A student's creativity. Typical students in grades 4 and 5 will work more closely with their environment and will often choose projects depicting objects that are familiar. Older students should be encouraged to use their immagination. A little "brain-storming" may aid the process.

The instructor has to consider another aspect, mathematics. The task of designing a CAD drawing with Logo can involve many calculations using ratio and proportion and often some trigonometric knowledge. The instructor must decide whether the CAD project is an exercise in designing, programming, or mathematics. Further explanation of this question follows in the section titled Mathematics and Measurement.

Linear CAD

What I will describe to you as programming in a linear fashion means beginning at one point on the turtle screen and proceeding with one continuous line until the drawing is complete and the turtle has returned to his starting position. This form of programming is often used by students who are not familiar with the Logo Editor or by students who see things not as parts of a whole but in a linear fashion. (See figure 1.) Notice that although the student called on procedures for doors and windows, the rooms are not separate parts of the whole floor plan. This

Linear CAD is difficult to debug and modify and may be suitable only for simple designs or students who are just beginning their Logo experience.

Modular CAD

What I will call modular CAD allows a great deal of flexibility and opens up many lessons in mathematics. Let us consider a floor plan for a house. By designing one room at a time and putting the whole floor plan together at the end, the student must have a clear knowledge of the sizes shapes, and requirements of the rooms. This approach also requires careful use of the Logo Editor. Procedures within procedures and may involve many layers. This allows the student to include parts in the drawing which are not connected to perimeters (a free-standing fireplace or spiral staircase). Students using a modular approach will be more inclined to use shapes other than rectangles and to add color for emphasis. (See figure 2.)

Mathematics and Measurement

The mathematics involved in any CAD drawing requires a knowledge of ratio and proportion. A convenient scale must be determined in advance to allow the drawing to fit on the turtle screen. In addition, CAD tools for making doors, windows, stairs, and the like must be designed to take inputs so that they can be used with students who may use different scales.

Students can develop their own CAD tools but much help may be necessary from the instructor for middle school students. Tools that require circular drawings may involve a discussion on radian measure as well as circumferences and tangents. Providing a wide variety of CAD tools for the students will maintain enthusiasm and allow the student to concentrate on the finished drawing.

A circular transparent protractor will become a convenient device for determining angles for students who may not have been taught about adjacent angles. This author feels that such protractors should be general issue for any Logo activities that involve depicting real objects.

Although students may choose any convenient size for their CAD drawings, it may be helpful for students to measure common place objects (doors, windows, bathroom fixtures and the like for floor plans) so that their drawings conform more closely to what makes sense and what may be required structurally if the design were to become a

construction plan.

Young students have difficulty measuring. Choosing a large figure to depict and using a student's foot as a measuring unit is most helpful. Students can practice pacing off the perimeter of their own home by placing one heel to the toe of the other foot. Round off or disregard any fractional "feet". Often the turtle step can equal one "people step".

Completing the Project

What follow is an abbreviated outline for undertaking this CAD project at two levels. The first level is appropriate for 4th or 5th grade students with some knowledge of Logo. The sections enclosed in << >> would be substituted for cognitively and mathematically more advanced students.

1) Present the problem. The instructor should provide guidelines for choosing a project with concern for the knowledge of Logo and mathematical abilities of the students.

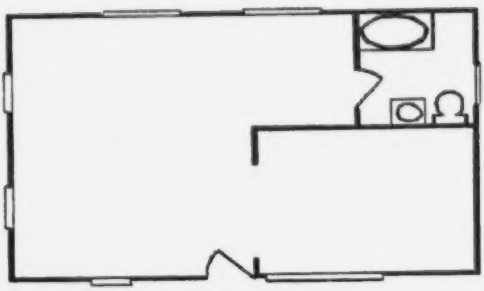
2) Have the students draw a rough picture using graph paper for their project. << Choosing an appropriate scale and drawing the rough picture to a scale that will fit on the turtle screen is adviseable.>>

3) Have the students write a list of what has to be constructed and order the list sequentially. << A Logo outline for a modular approach would involve listing the parts of the design and the procedure names of the tools or sub-proceddures necessary for each part.>>

4) Move the turtle to an appropriate starting point and begin to draw the design in Command mode. << Advanced students will want to begin a super-procedure and sub-procedures for the drawing.>> Incorporate the CAD tools in your drawing as necessary.

5) De-bug, modify, and print out with the appropriate hardware.

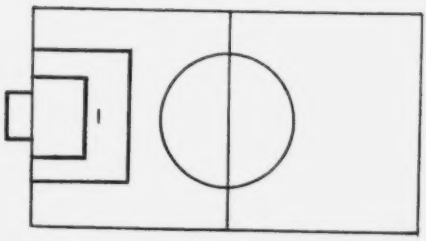
The CAD project need not be limited to the computer or mathematics curriculum. Any curricular topic where students are asked to do accompanying art work is a suitable place to begin.



```
TO FLOOR.PLAN
PERIMETER LIVING.ROOM
BATH
END

TO PERIMETER
PU SETPOS [0 -40] PD HT
LT 90 DOORR 25 WALL 40
WIN 20 WALL 40 RT 90
WALL 30 WIN 20 WALL 40
WIN 20 WALL 30 RT 90
WALL 50 WIN 40 WALL 30
WIN 40 WALL 80 RT 90
WALL 30 WIN 20 WALL 90
RT 90 WALL 50 WIN 60
WALL 5
END
```

```
TO TOILET
HT FD 5 RT 90 FD 5 LT 150
ARCR 7 300 LT 150 FD 5
RT 90 FD 5 RT 90 FD 17
END

TO BATH
PU BK 60 LT 90 WALL 10
DOORR 20 WALL 30 RT 90
BATHTUB 40 RT 90 PU
FD 60 RT 180 PD TOILET
FD 22 RT 90 SINK 18 HT
END

TO LIVING.ROOM
RT 90 WALL 10 PU FD 50
PD WALL 20 RT 90 WALL 113
END
```

Figure 2.



```
TO SOCCER.FIELD
PU SETPOS [-100 -80] PD
FD 90 RT 90 FD 75 RT 90
FD 90 BK 90 LT 90 FD 75
RT 90 FD 90 RT 90 FD 150
RT 90 FD 15 RT 90 FD 40
LT 90 FD 60 LT 90 FD 40
LT 90 FD 12 LT 90 FD 16
RT 90 FD 35 RT 90 FD 16
RT 90 FD 7 LT 90 FD 8
RT 90 FD 20 RT 90 FD 8
PU FD 24 RT 90 FD 7 PD
FD 6 BK 3 LT 90 PU FD 75
LT 90 PD ARCL 25 360 HT
END
```

Figure 1.

Creative Use of Logo with Special Student Populations: Using Atari Logo in the Primary Grades at Bailey's Elementary School with ESL Students

Steven M. Adleberg
Bailey's Elementary School
A Fairfax County Public School

I arrived at Bailey's Elementary School two years ago. One's first impression of Bailey's is an accurate and lasting one. The glowing faces of the children represent seemingly every country in the world. It's like stepping into a children's "United Nations" building. And in fact, most of the nations of the world are represented in the student population. "We've got countries you've never even heard of," Fairfax County Supervisor Tom Davis relates. The Bailey's community, after Arlington, is the second largest "new immigrant" community in Northern Virginia. South and Central Americans, Afghans, Turks, Greeks, Pakistanis, Iranians, Lebanese, Koreans, Vietnamese, Thais, and Scandanavians are just a sampling of the countries represented by Bailey's students. Children (and/or their parents) come from over 40 countries, speaking over 50 languages and dialects. Recently, a kindergarten teacher pulled two shy but giggling boys into the principal's office, reporting that "this one stuck his pencil in this one's arm." One child had to be disciplined in Spanish; the other in Urdu.

Carrying this impression and information, I eagerly commenced my new role at Bailey's as Special Projects Teacher for Math and Computers. I was eager to develop a Logo computer language curriculum, especially for the primary grades. It should be noted at this point that I had started developing a Logo curriculum three years earlier in the school environs of homogenous upper middle class communities in the northwest potions of Fairfax County. When I first introduced Logo vocabulary to kindergarten, first, and even second grade students at Bailey's, most children could not pronounce the word "Forward" much less know their "Right" from their "Left." Playing "Logo Turtle Says," the students gradually understood and integrated the basic commands (**FD, BK, RT, LT, CS, HOME**) into their reading and speaking vocabulary and into their body movement. The next step was to introduce the Logo light turtle so that with their newly integrated Logo vocabulary the children would proceed to create different shapes and designs on the computer. I realized quickly, however, that most students were getting bogged down when they did get on the computer. The turtle wasn't responding the way they (or I) had hoped. The cause, I surmised, was a language problem.

In order to maintain the children's energy and enthusiasm for the Logo turtle, I felt the need to alter and accelerate the Logo learning process. I forsaw the students' frustration developing if alternative Logo teaching strategies were not employed. As an introductory series of lessons, I thought that instead of the children teaching the turtle to do things, at least at first, the turtle could teach the children. I began by creating single keystroke procedures for the above Logo commands. In all the single keystroke procedures that I created for the children I slowed the turtle's action (using the **SETSP and WAIT** commands) so that students could actually trace (with their fingers) the line or shape being drawn on the monitor. With descriptive text appearing in the lower part of the **Splitscreen (SS)**, the turtle acted as teacher, model, and guide. We discussed the special attributes of each geometric shape that was accessed simply by entering its name after the Logo question mark prompt. The students were enthralled, drawing squares, rectangles, triangles, circles, trapezoids, parallelograms, pentagons, and more. They drew with their fingers on the monitor screen, with their crayons, pencils, and paint brushes on paper, with popsicle sticks arranged on the table, and with clay on the clayboards. We talked about the shapes. The children generated imaginative explanations of the lines, shapes, and their combinations and verbally shared those images (some real and some very unreal) with me and with each other before the close of each class. On a class chart, we kept an organic list of every new shape that was introduced with a printer generated Logo graphic beside the name of each new procedure. The excitement contiued to grow each day with the children entering the lab to discover and learn new geometric shapes. They used the **Penup (PU), Pendown (PD), and Pen Erase (PE)** commands to draw more than one shape on the screen at a time.

Each day students entered the lab to find a new procedure on the chart. They had no idea what the word meant or what the turtle would do until they carefully typed in the name of the new procedure and hit the **RETURN** key. They saw the word in print; they typed it, letter by letter, on the keyboard; and they finally watched in rapt amazement as the turtle defined it before their excited eyes. They reinforced the word and its concrete meaning by returning to their tables and using the above mentioned manipulatives, calling out the shape name again as they completed their creation. The turtle was such an effective language arts and mathematics teacher, model, and guide that most students became linguistically prepared to turn the tables and were ready to begin teaching the turtle.

In Fairfax County Schools, the teaching of Logo (on-line and off-line) is not intended to be a curriculum in itself; rather, Logo activities are to be used to meet Program of Studies objectives. Some Fairfax county Public Schools Level A, B-1, B-2 ESL Instructional Objectives (**Study Skills and Functional Literacy Skills**) relate directly to Logo computer language learning:

-Students follow one- and two-step oral and written directions; give one- and two-step directions orally and in writing.

-Students follow multi-step oral and written directions; give multi-step directions orally and in writing.

-Students classify objects, pictures, and words by one or more of the following attributes: color, number shape, size, or function.

-Students demonstrate understanding of basic Logo language vocabulary and its meanings relating to the English language.

-Students tell stories about pictures or other stimuli, relating important events in a logical sequence.

-Students tell their own experiences before a group, presenting important information in order.

The Deparment of Instructional Services has made the instruction and exercise of students' critical thinking skills a county-wide goal. Higher level questioning strategies as outlined in Bloom's Taxonomy are emphasized when working with ESL students in the Logo language computer lab.

Fantavision in Education
Scott Anderson
Freelance Programmer

Introduction

Fantavision is a special effects animation generator for the Apple. Animation traditionally involves extremes of action (key frames) to advance the plot. Smoothing the extremes are dozens of laboriously created in between or 'tween' frames. Fantavision does these tweens automatically, freeing the animator for creative work. Unlike Bit-mapped graphics, there are no restrictions on the size of an animated object. An object is a polygon and is represented by its vertices. Each vertex has an X and a Y value. A square, no matter what size, can be represented by just eight bytes. Since only the key frames are stored (and they are small), a lot of animation can be crammed into memory.

Animation can be a boon to educators. The blackboard is inadequate to explain many concepts involving growth or motion. These same subjects can become simple and intuitive when animated. Since Fantavision animations are interactive, the students can proceed at their own pace, repeating any concept that eludes them. This talk focuses on some of the applications of interactive animation to education. We will also explore some animation tricks for educators who wish to create their own movies.

The following is an outline of some of the topics to be covered in this talk.

Simple tricks with Fantavision

Rope trick: Use three frames to bounce a ball on a rope.
Bouncing ball: Pay attention to the parabola for realism.
Explosion: One dot becomes sixteen, then poof!
Text Rotate: With only two frames, a 3-D rotation.
3-D stills with tweening: In trace mode, create dimensional objects.
Paste a tween: To smooth animation or create oddities.

Animation as a subject

Motion: Illustrate the concepts of animation.
Tweening: Illustrate interpolation and transformation.

Animation as a creative tool

Animate stories: Create movies to send to friends.
Special effects: Experiment with lightning mode and other goodies.
Titling: Animate text or add titles to your VCR movies.

Using Language hooks

Title Pages: Create animated titles for your programs.
Enhance programs: Add special effects to your BASIC programs.
Create games: Make interactive educational adventures.

Creating your own

Video as source: Beta is best to find the extremes of motion.
Multiple exposures: Sports magazines often have sequential stills.
Muybridge: Eadweard Muybridge spent years analyzing motion.
Imagination: Create a fantasy world.

Animation techniques

Straight ahead: Animate from frame to frame with no plot.
Pose to Pose: Carefully plot the action with key frames.
Squash and stretch: Use the goodies menu to add feeling to a motion.
Anticipation: Build a little suspense to prepare the viewer.
Arcs: Most body motions take a circular path, so add key frames.
Staging: Keep it simple for maximum clarity.
Inertia: An object in motion tends to stay in motion.
Secondary action: Add realism with extra activity.
Exaggeration: Enhance an attribute for effect.

Presentations

Stop motion: Use the space bar for still frames.
Speed change: Use the number keys to increase or decrease the speed.
Direction change: Go back to a previous movie with the arrow keys.
Skip movie: Press the Return key to skip to the next movie.

Examples

Several Fantavision movies will demonstrate concepts in biology, mathematics, physics, art and english.

References:

Computers and Creativity, Part II
by David D. Thornburg
A+ Magazine, January 1987
Ziff-Davis Publishing, New York, NY

Fantavision Trickery
by Scott Anderson
II Computing, June/July 1986
Antic Publishing, San Francisco, CA

The Human Figure in Motion
by Eadweard Muybridge
Dover Publications, New York, NY

Disney Animation
by Frank Thomas & Ollie Johnston
Abbeville Press, New York, NY

LOGO ACTIVITIES AND STRATEGIES IN THE KINDERGARTEN

SUZANNE L. BAILEY
FAIRFAX COUNTY PUBLIC SCHOOLS
FAIRFAX, VIRGINIA

With the recent influx of added subjects like Logo in the elementary school curriculum, teachers are finding it difficult to adequately cover all the subject areas during the school day. My solution to this problem is the integration of Logo into the various subject areas. My presentation will address integrated Logo activities and strategies I have designed and implemented in my Kindergarten classroom.

It is absolutely ludicrous to view Logo instruction as an entity in itself without recognizing its affects on other curriculum areas. Logo has become an exciting tool which my students use in order to learn math concepts and skills, as well as skills and concepts in other areas. Logo has become a tool of instruction for me whereby I can simultaneously teach skills in three subject areas in one lesson. For instance, "The Great Escape," a Logo activity I have designed, teaches mapping skills, while simultaneously reinforces safe fire drill procedures and movement skills.

Logo not only involves all curriculum areas, but I have found that it affects the cognitive, psycho-motor, and affects socio-emotional areas of development. In the area of psycho-motor development, young children frequently need to use their bodies to learn and subsequently internalize abstract concepts. By mimicking the movement of the turtle, children can begin to understand the concepts of directionality and laterality. Young children are active learners and my students eagerly imitate the turtle's movements through "The Turtle Magic Rap Song." This rap song is a movement oriented song which teaches the Logo commands, as well as the movement of the turtle.

For young children, success in the school environment is essential in building a health attitude towards learning and building self-esteem. These components are of utmost importance in the socio-emotional development of a child. Logo is an excellent way in which a child can expxerience success on his individual ability level. On the other hand, it fosters a respect for individual ideas, as well as cooperative teamwork in finding a solution to a problem.

Teachers of young children have an exciting and challenging responsibility to afford their students with opportunities which will lead to cognitive development. Once again, Logo is an excellent tool which facilitates intellectual development. I have discovered that reading

readiness skills are easily taught by utilizing Logo in the classroom. When using Logo, children actively use their languages and as a result, their vocabulary and language skills are developed and honed. My students are excited and eager to write about the exploits of the turtle. "Turtle Tales," a home reading incentive program, has also been very successful in my classroom.

Of course, a five year old child finds the adventures of the turtle fun and exciting, but parents do not always see the merits of Logo. My "Turtle Tales" fosters parents participation and an understanding of Logo and its benefits. Other strategies to involve parents will also be presented.

A successful Logo program should be integrated into the daily task of learning. For this reason, a teacher should "logorize" their classrooms and their children's thinking. My students' problem solving abilities are becoming more refined due to this total immersion in a Logo environment.

Teachers of young children know that children have a relentless curiosity. This curiosity unfolds as they manipulate materials to find out about the world around them. As with hands-on experiences, children learn when their experiences are made meaningful. For this reason, learning must be a process whereby all the child's senses and processes are utilized. Logo certainly meets this criteria for learning.

This presentation will focus on the activities and strategies I have used that develop all areas of a young child's development. I will share on-line and off-line activities in various curriculum areas like reading readiness, music, movement and art. I hope that you will join me for some turtle magic!

Geometry through LOGO for Teachers in the Middle Grades*

Hunter Ballew, Professor of Mathematics Education
and Director, Center for Mathematics and Science Education
School of Education, University of North Carolina
Chapel Hill, North Carolina

Introduction

LOGO is an authoring system. It took me some time in working with LOGO to realize that. But that is exactly what it is. LOGO enables the teacher to design lessons that combine geometric figures with text so that students can learn geometric concepts in an interactive way. LOGO may not be as easy to use as some other "authoring systems," but it is quite powerful.

This paper will describe a general approach to teaching teachers about the uses of LOGO in teaching selected geometric concepts in the middle grades, and will discuss some of the features of LOGO that make it a powerful tool for designing lessons and providing instruction in geometric concepts.

General Approach to In-service Education in LOGO

Students often yearn for all information, regulations, and requirements to be laid out clearly for them in the beginning. "Just tell us what to do," they beg, "and we will do it." However, with all the commands, the techniques, and the power of LOGO, too much information in advance of action can be intimidating. Learning LOGO is like learning tennis--you can't learn all the rules and all the moves and all the strategies of tennis before you hit the first ball, and you can't learn all about LOGO before the first keystroke.

My experience is that it is better to resist this desire for an overload of preliminary information, and to start with a few simple keystrokes that make things happen on the screen. This is probably even more true for teachers who come to us in in-service classes than it is for the usual undergraduate student. Teachers who come back as students are adult learners who bring to their classes a wealth of experience and knowledge. These adult learners are likely to constantly be comparing what the instructor is

*The materials referred to in this paper are being developed as part of a grant from Academic Information Systems of IBM.

telling them to what they have already experienced back in their own classrooms. For this reason, it is better to follow these steps in teaching teachers to use LOGO as an instructional tool:

1. Prepare a few interactive lessons on disk as illustrations of some of the ways LOGO can be used to teach geometric concepts. This is to get teachers acquainted with the art of the possible.

2. For those who have no experience with LOGO, prepare activities that will teach a few basic LOGO commands. Other commands can be learned as the need for them arises. For example, if a student is in a situation requiring an action to be repeated a number of times, this seems a good time for the student to learn the power of the REPEAT command.

3. These lessons and activities are not intended to be completely self-instructional. An instructor should be available to supply information as needed and to make suggestions. For example, if someone is typing the same command repeatedly, or similar commands in sequence, the instructor might take advantage of the opportunity to explain the usefulness of the F3 key. In addition to the instructor, one or more LOGO reference books should also be available to the students at all times.

4. Ask all students to keep logs of commands they find useful, together with an explanation of how to use the commands and a description of what each command causes to happen. This should be done with even the simplest commands until they become thoroughly familiar to the student.

5. Assign a project for each student to complete by the end of the course. This project should require students to choose a geometric concept from the grade level in which they are most interested, and then to prepare an interactive lesson on disk to teach that concept. The purpose of this project is to extend and cement the familiarity of the student with the commands and potential of LOGO. In fulfilling this purpose the student should draw upon the knowledge found in reference books, school textbooks, their classmates, their instructor, their own notes, and on experience gained from trial and re-trial of various ideas on the screen. Design and development of the project will provide students with a facility in the use of LOGO and with ideas about how LOGO can be used in their classrooms.

Motivation, thus achieved through LOGO, inspires experiential and more permanent geometric learning that goes far beyond memorization of commands and learning of programming procedures. This kind of instruction entices

the participants into active problem solving which is the antithesis of the inert knowledge Alfred North Whitehead warned against long ago as a debilitating ingredient in school learning.

LOGO Features Useful for Design of Lessons

Color: It is impossible, in my judgment, to over-emphasize the value of color for interactive lessons on the computer. Color can be used to instructional advantage in the following ways:

1. Emphasis. A word being defined might be printed in a color different from the other text on the screen. For example, in defining supplementary angles, you might want to put the word supplementary in a different color. Any words or phrases you wish to call attention to could be put in a different color to make them stand out.

2. Labelling. When several angles appear in one drawing, young learners often have difficulty with labelling schemes. They have trouble discriminating among the various angles in the drawing. Angles being referred to in a drawing can be made to stand out by using different colors. For example, suppose you wanted to call attention to a particular pair of corresponding angles in a drawing of a pair of parallel lines cut by a transversal. You could label those angles in red while all the other angles in the drawing are labelled in green.

3. Classification. Figures of one type can be drawn in one color while figures of another type can be drawn in a different color.

4. Interaction. In a lesson on the computer, confusion can be avoided by using different colors for instructions to the student and input by the student. For example, general instructions to the learner could be printed in gold, precise input the learner is being asked by the program to type could be printed in red, and the actual input of the learner could be printed in green.

5. Attractiveness. On the text screen, sixteen different colors of text can be used. On the graphics screen sixteen different background colors can be used, and six different colors can be used for drawings and text color. A slight handicap to the lesson designer occurs with LOGO in that only three of these pen colors and text colors can be used at one time on the graphics screen. If you select Palette 1, you can get lines and text in green, red, and gold on the

graphics screen at the same time. If you select Palette 0, you can get lines and text in blue, magenta, and white. However, with these color combinations, many attractive screens can be designed. In the near future, it is expected that systems with more possible colors will become commonly available.

Position: Text and diagrams can be placed in different locations all around the screen. This provides a great deal of flexibility for the lesson designer, both for attractiveness and for the effective presentation of information, diagrams, and exercises. For example, instructions you want to keep through several exercises can be placed at the top of the screen while a diagram remains in the middle, and exercises are changed at the bottom by the CLEARTEXT command.

Movement: Drawings already on the screen can be added to while the learner watches. For example, medians might be added to an existing triangle. Movement of the turtle can be slowed down so the learner can observe a sequence of events. Angle interiors can be shaded in a way that shows graphically the rotation of the turtle in forming angles. Closed figures can be filled with color by a moving line both to call attention to a particular figure and for attractiveness.

References

Avery, R.R. (1985) LOGO and the IBM PC. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Burrowes, S., Tobias, J., Short, J., and Lough, T., with Riordon, T.P. (1985) Exploring IBM LOGO: A Guide for Adults. New York: Holt, Rinehart and Winston.

IBM. (1983) IBM Personal Computer LOGO Reference. Boca Raton, Florida: IBM, 1983.

IBM. (1983) LOGO: Programming with Turtle Graphics. Boca Raton, Florida: IBM.

Watt, D. (1986) Learning with IBM LOGO. New York: McGraw-Hill Book Company.

THE ENFI PROJECT

Trent Batson, Ph.D
Gallaudet University
Washington, D. C. 20002

"Even at grade 11, fewer than one-fourth of the students performed adequately on writing tasks involving skills required for success in academic studies, business, or the professions. In general, American students can write at a minimal level, but cannot express themselves well enough to ensure that their writing will accomplish the intended purpose." -- The Nation's Report Card, November, 1986 (Educational Testing Service).

For three years at Gallaudet University, we have successfully employed a "total immersion" in writing project that confronts the writing problems of incoming freshman students. Believing this computer-based program to be appropriate only for deaf students, we were surprised to find interest among educators at other universities.

This total immersion method -- English Natural Form Instruction (ENFI) -- requires a local-area network of more or less 12 micro-computers set up in a classroom so teacher and students can talk to each other in writing.

Because of our encouraging results and outside interest, therefore, we've begun to expand the Project beyond Gallaudet. It's attractiveness for hearing students lies in its ability to strengthen classroom practice in:

* the writing process,
* collaborative writing,
* writing-across-the-curriculum and
* connecting speaking and writing,

which are methods or theories for teaching writing dominant in the profession now.

Though teachers report using these methods, the Educational Testing Service believes they are employed superficially (at least in high school). "It is the how that should be at the heart of the movement to reform writing instruction," they believe, a plea for putting these theories into practice, not just giving them lip service.

At the very least, the ENFI method provides a highly manipulable shared working space, allowing the teacher to work with the students' text as it is generated. At most, it achieves the goals of a total-immersion approach to language instruction: significant improvement in a short time.

THE THEORY

Behind the ENFI Project is this idea: by making writing more like speaking, as we do on our computer network, students can build on their oral fluency, which is normally more developed than their written fluency.

In most composition classes, teachers discuss ideas in class and ask the students to write about those ideas for homework. Apparently they assume that if students can talk about an idea, then they can write about that idea. Even in this day of the writing process, they seem to see writing as merely putting down what is in your head. This assumption is almost certainly fallacious. In a speaking situation, there are numerous cues and supports that help you get through an idea; in writing, you're usually on your own. The two skills are very different.

So, a couple of crucial questions arise:

1. How do we create some bridging techniques for students so that there is not such a lag between talking and writing, between writing and feedback, and between feedback and revision?

2. How do we bring students' written fluency up to the same level as their oral fluency? Or, in other words, how do we bring to bear the techniques used to develop oral fluency in writing?

To get the ideas in class down on paper while the ideas are "hot," teachers do ask students to write in class, but often the shift in context -- loss of the social interplay, loss of cues, loss of the motivation -- is almost as severe as if the writing was done for homework.

Written composition, after all, is performance. It is a complicated creative process bound by strict rules of form; writing music or choreographing dance are similar skills. We shouldn't expect students to perform such a process on their own; ideally, the teacher would constantly interact with the student as the student writes his or her composition.

Interacting in this way with 20 or 30 students as they write on paper is obviously impossible, but writing on our computer network makes it more nearly possible, at least in the early stage of the writing process. (Though we limit the classroom to 12 computers, we are able to handle a full-size class by having more than one student per computer or dividing the class and having only half the class working on the computers at a time).

First, because we interact in writing for our discussion of

ideas, we provide a way for students to experience writing that is less formal and more conversational. Interacting in speech is how they developed their oral fluency; interacting in writing has the same salutary effect.

Second, because we move the students from the discussion to the first stages of the writing process -- all in writing -- we provide a bridge between speaking and writing, a bridge that never existed before.

For students who are not already competent writers, the "ENFI bridge" provides motivation, resulting in more attention to the writing task. For students who are competent, the ENFI bridge encourages including more speech-like elements in their formal writing, such as including personal elements or providing the "mowing long grass" feature of referring back to ideas already mentioned. (Elbow, Nov., 1985) They observe the teacher writing, something they almost never do in a traditional class, and they also see their own writing mixed in with that of the other students, allowing them to see how well they do. This pattern of fitting ourselves in with others in a speech community occurs in speech all the time, of course, and so speaking seems more "natural," more a part of us, more something we identify with. Writing doesn't seem so natural only because we've not had the same free-interactive experience in writing we've had in speaking.

OUR TARGET

Our Project is aimed mainly at composition teachers in college -- and their students -- but we also work with educational computing people.

Schools are stuck now with the idea that computers will teach, one-on-one, using educational software. Such a notion is a blind alley because software is not truly interactive but inter-promptive and the social dimension of writing is lost. In challenging this "presumption" of "a lone individual, working in at least temporary isolation," several researchers (Levin, et.al., 1985) encourage the use of collaborative writing on computers because it "provides social resources to confront the blank screen."

Students seem to learn best by interacting with a live person and no software program can simulate that, at least not yet. For computers to move to center-stage in the classroom, they have to be part of a live interaction, participating in and aiding instruction, not supplanting it.

Computers networked as managers of classroom interaction have a deep and positive effect on education. They open a new channel of communication between teacher and students that is less

daunting in many ways than face-to-face communication. The traditional role of the teacher as unerring authority is subtly altered. Ideally, teacher and students become partners in co-learning.

We find that ENFI has these results:

1. Students, learning to use writing as they do speech, start to use writing in a greater variety of ways -- for example, to work through ideas. They gain in fluency because of the heavy dose of interactive writing over a semester or year. Writing seems less mysterious, less unique to English class, more their own.

2. Teachers find new ways to use current theories about composition. They experiment with a new method that gets them untracked and extends their boundaries.

3. Teachers and administrators see demonstrated a new model of computer use in the classroom, not only for writing instruction, but for other uses as well.

Gallaudet now has five networked classrooms which serve about 200 students each semester. ENFI is a standard method on campus supported by a cadre of loyal teachers. Students taking classes on the ENFI network do as well or better than students in "traditional" classes and both students and teachers are satisfied with the method.

This year, ENFI networks at Northern Virgina Community College (NVCC) and the University of Minnesota are being used with regular college students for the first time. Teachers at NVCC taught several freshman composition classes in the fall. At Minnesota, they're just beginning using ENFI this winter term. Because our Project is so new at these other sites, we have no "hard" results to report. However, at NVCC, teachers report that both they and their students liked the system in the fall.

******

Trent Batson, Ph.D, Director, ENFI Project
Gallaudet University
Washington, D. C. 20002

(202)651-5494
(202)651-5000 ext. 4496 (tty)

One Computer --Thirty Kids--and You Want Me to Use Logo??

Jane Bauserman, Norfolk Public Schools
Norfolk, VA

Dianne Lawrence, Consortium for Interactive Instruction
Norfolk, VA

Introduction

There are several approaches to using Logo in the classroom when only one computer is available. The teacher should act as a guide or facilitator whether working with an entire class, a pair of students, or a small group. The PROCESS not the product is the focus of the Logo philosophy. Exploration and discovery take place as students learn Logo commands.

The purpose of this paper is to provide the teacher with practical ideas for organizing a classroom. Many of these techniques will not be new. They are just variations of the management techniques that many teachers have already utilized.

Whole group

When working with a whole group it is very important that all students be able to see the monitor. By connecting an RF modulator to a color television set a large screen monitor is created that can be seen by an entire class. This provides the teacher with an invaluable tool when introducing new Logo commands. The teacher may demonstrate the command and then allow individual students to explore as the entire class observes and makes suggestions. This time should be kept purposely short to allow students the pleasure of discovering many of the "what-if's" on their own.

Logo command exploration can be aided by transparent mazes and dot-to-dots. The large screen monitor enhances a brief demonstration by the teacher before the aids are explored by the students. Visual demonstrations can limit the number of "What am I suppose to do now?" questions.

The large screen allows a "showcase" for students to demonstrate discoveries to be applauded by their peers. It also provides a forum for group problem solving of "bugs" that are approaching students' frustration level. This sharing of achievement or difficulty leads to extensions of the present project and inspirations for new projects for the entire class. A by-product of this process is the increased self-confidence gained by the students as they control the machine and the Logo environment. The ability to accept and recognize the ideas and knowledge of peers is encourged in this non-threatening approach to problem solving. This is the Logo philosophy in action.

Student Pairs

Working in pairs provides another approach to exploring Logo. Pairing students of differing abilities fosters shared learning. As in other areas of instruction, this has to be done with care. The classroom teacher has extensive experience with the students and can use a variety of methods in determining which students work together most effectively. It is highly recommended that students be allowed the opportunity of learning to work with a variety of partners. This provides different approaches and perspectives on solving a problem.

Managing Movement

Managing the movement of students from desk to computer should be planned carefully. If students are aware of the expected process, classroom disruption will be minimized. Once a system is put into practice, sufficient time should be allowed for the system to become the "norm". Expect problems and be pleasantly surprised when they are minimal. After the system has been tested for approximately two weeks, evaluate its effectiveness and be willing to revise as necessary.

Keep in mind fifteen to thirty minutes is recommended as an appropriate length of work time at the computer. Of course this time can vary with the age, attention span and type of Logo project. A timer can be used to help the students keep track of the time and prevent clock watching.

Some methods of management include flip cards, lists, and time-on charts.

Flip cards: A teacher can make cards displaying each student pair or group. The teacher can then select the working order of the students by the placement of the cards on a metal ring. As the first pair/group finishes, the card is flipped over and the next pair/group is quietly notified. When the cycle is finished a stop sign appears and directs the last pair/group to quietly notify the teacher.

Lists: A program such as MECC's "Labels, Letters and Lists" may be used to generate a listing of students as a check sheet. As the turn is completed the pair/group is checked off the list and the next students notified.

Time-On Chart: A time-on chart allows students to sign up for the specific time they would like to work at the computer during the week or month. Some slots are reserved for students that missed their designated time because of unexpected interruptions or illness.

Monitoring Progress

Monitoring student progress can be accomplished by use of disks and/or student logs. As students complete their turn, work should be saved to the data disk. Upon return to the student's desk, a maximum of five minutes should be allowed for recording in the student's Logo log. Completed procedures and the setting of goals for the next session should be included. The log should be used for reference by student and teacher. The log provides a concrete place for Logo ideas, successes, and goals to be recorded. Students should be encouraged to write briefly about their feelings during the work session. The log will help students to remember what has been and what needs to be accomplished. The teacher should review the log to monitor the student's individual and group progress.

Small Group

Problem solving in Logo lends itself well to small group participation. The group may include a team leader, typist, reader, encourager, and syntax "command-er". The assignment of duties to each student in the group makes the student an equal "owner" of the project and encourages active participation. These roles may be determined by teacher or students. Although each group member has a specific duty, all are responsible for creative thinking. A good deal of interaction must be expected if the group is to function effectively.

Students need planning time and computer time when working on group projects. Designated off-computer planning time maximizes the efficiency of the time students spend at the computer.

Two approaches to group programming can be labeled "soft" and structured. "Soft" programmers are those who use the computer as an artist uses canvas and paint. A dab here and there with feeling creates the finished project. Programmers who take the structured or "top-down" method decide upon the finished product and then develop the necessary procedures.

Structured programming via group effort can be achieved by requiring the group to sketch the finished product in their journals. The group must then analyze the superprocedure and decide upon the subprocedures needed. Mini teams will concentrate on creating the needed parts and a "glue" team will be in charge of merging the subprocedures into the finished project. Members of the "glue" team are not excluded from the other phases of the group effort.

Summary

It is very important to remember there is not one best way to organize every classroom to include Logo. Each group of students, teacher style, subject matter and software used will effect the way a classroom is organized. But while you are still organizing your own thoughts and ideas about what method will work best for you and your students, allow yourself the freedom to try several different ways. No matter what approach you select it will take time to make things work smoothly. Be patient with yourself and your students. Talk with other teachers, and ask your students if they have any suggestions.

With the exception of the whole group method, whatever scheduling system is used should require MINIMAL TEACHER ASSISTANCE. The system should minimize classroom disruptions and maximize computer usage.

BIBLIOGRAPHY

Bear, George G., Microcomputers and School Effectiveness, Educational Technology, January, 1984

Study Examines How Teachers Organize Computer Use, Electronic Learning, February, 1985

Torgerson, Shirley, Classroom Management for Logo, The Computing Teacher, December 1983

Turkle, Sherry, The Second Self, Computers and the Human Spirit, Simon & Schuster, Inc. New York, 1984

Creating a Learning Environment for Teachers

Alison Birch
Terrapin, Inc.
222 Third Street
Cambridge, MA 02142
© 1987 Alison Birch

Providing a creative learning environment is central to the Logo philosophy. However, Logo is not a philosophy in and of itself, but a tool that supports a more general educational philosophy—one that promotes exploration and discovery.

Much research has been done concerning transfer of learning in children. Can students who use Logo apply their learning to other areas? A far more interesting question for me is whether or not *teachers* transfer their learning of Logo to their own classrooms and their own teaching. Teacher training in Logo can all too often consist of lectures on how teachers should provide open environments for children to play with and explore Logo. But it is also essential for teachers to experience that same discovery based learning in order to understand how their students feel and to know *for themselves* what a powerful tool Logo can be.

A Program for Learning

The SummerMath for Teachers program at Mount Holyoke College, South Hadley, MA provides an active learning environment for teachers. This presentation gives an overview of a two-week program for secondary math teachers in which I have been an instructor for the past two years. The program involves three different learning components—cognitive process instruction, physical activity, and computer lab. Each of these components provides an opportunity for teachers to become students and learners, not only in their own discipline but in new areas as well.

The fundamental philosophy of the SummerMath for Teachers program is *constructivism*—giving students opportunities to become actively involved in their own learning and in constructing their own understanding of concepts. The teachers learn constructivist methods by being students in a constructivist learning environment.

Teachers who have been taught by the traditional lecture/demonstration method usually use the same method in their own teaching. They "cover" the class material by methodically going through the textbook. The students learn to memorize algorithms and procedures for solving problems like those in their book. They don't necessarily understand a given concept, but have memorized a process. They haven't learned to think mathematically.

In a constructivist classroom, students become actively involved in their own learning. The teacher is no longer simply a person who gives information, but someone who creates activities that encourage students to ask questions. Through pair problem-solving and group work, students have a chance to share their ideas and strategies with others. This doesn't mean to say that lectures and demonstrations are not effective teaching methods. They are effective for imparting factual information. But, for learning concepts and for helping to make abstract ideas more concrete, exploration and discovery are much more powerful.

The first component of the SummerMath for Teachers program is a morning of cognitive process instruction. This approach to teaching emphasizes the role of students as active learners and problem solvers. The teachers are given the opportunity to reflect on their own problem solving process just as they will emphasize this method with their students. The program leaders are committed to modeling the kind of teaching they advocate and are encouraging in the teachers.

Exercising the Body as Well as the Mind

The second component of the program involves physical activity—something the teachers need after a morning in the classroom. Each teacher spends one week in a dance class and another week in a tennis class. Although there are several groans about this at first, most of the teachers find the experience very enjoyable. One teacher from last summer made the following observation about this part of the program:

> "Dance is the same type of piecing together as programming. We learned our dance combination one step at a time and then pieced it together step after step. We're programming by creating procedures to do simple things and then piecing the procedures together. Now I see the connection between the two seemingly different classes."

For many, it is the dance class or tennis class that helps them see what it is really like to be a student again and to learn something completely new.

Logo in the Classroom

In the SummerMath for Teachers program, Logo goes along nicely with the constructivist approach. The Logo component has basically two purposes: to provide teachers with an environment in which they will use Logo as a tool for developing new understandings of mathematics and to create situations in which teachers will be encouraged to see mathematics in a new way.

There are always plenty of teachers who want to *learn* Logo, and to some it is frustrating not to be taught everything. But the purpose is not to *teach* Logo, but to provide a model of the kind of teaching they will be doing with their students by actually having them *be* students in a Logo classroom.

For teachers who already use a lot of manipulatives in their classrooms, Logo is a natural addition to their teaching methods. However, for most high school teachers, the use of manipulatives is very infrequent if not, infact, completely non-existent.

In a Logo classroom, as in a constructivist classroom, the teacher creates learning opportunities or situations for the students. Creating a learning situation can be as open as giving students the time to explore their own project ideas or posing specific problems dealing with a particular concept. A classroom in which exploration and discovery are emphasized does not necessarily mean an unstructured classroom, but a flexible one.

One of the most interesting challenges for me as an instructor in this program is to develop problems that will be challenging to high school math teachers, but will not assume a prior knowledge of Logo programming. In addition, the problems need to be ones that will encourage further exploration by the learner.

For example, if you think starting with a SQUARE is too easy for people, then take away their FORWARD and RIGHT and ask them to draw a square using only the commands LEFT and BACK. Take something that is familiar and present it in an unfamiliar environment. A tic-tac-toe board, for example, is simple to draw on a piece of paper, but the same method for drawing this design with a pencil is not a very efficient method for Logo. By looking at familiar objects or problems in unfamiliar or different situations, you can begin to see patterns and relationships you never saw before. The following instruction shows one method for drawing the tic-tac-toe design in Logo:

```
REPEAT 4 [BK 20 FD 60 BK 20 RT 90]
```

To facilitate the sharing of ideas, the teachers work in pairs as they do in the morning class. As I watch the teachers enthusiastically working together to solve problems, I am always excited, but never surprised, at the amount of mathematics that goes on. Any teacher would be thrilled to see that kind of activity and involvement by students. At the end of the first class one summer, we got together to talk about whether anyone felt they learned something new mathematically. The teachers were silent. I kept thinking of my observations and wondered about the question, "What is it to *know* something?" Can you *know* something in one environment and not *know* it in another? Take Logo, for example, and a high school mathematics teacher. Certainly the teacher *knows* that the exterior angle of a regular triangle is 120°. But, in the Logo environment, that same teacher will attempt to draw an equilateral triangle by turning 60°—and will be surprised by the result.

```
REPEAT 3 [FD 50 RT 60]
```

Consider the Total Turtle Trip Theorem. How many teachers have actually discovered for themselves that the sum of the exterior angles of any regular (and non-regular) convex polygon is 360°? Sure they *know* it—they teach their students this fact every year—but they haven't actually experienced the excitement in discovering this fact for themselves.

We often ask the teachers to draw a regular five-pointed star in Logo. Most high school mathematics teachers will pull out pencil and paper and draw a complicated diagram involving regular pentagons, isosceles triangles and supplementary angles. If we then ask them to use what they learned from drawing the five-pointed star to draw a seven-pointed star, they will undoubtedly go back and take a closer look at that five-pointed star or find themselves involved in some tricky trigonometry.

```
REPEAT 7 [FD 50 RT 1080/7]
```

Encouraging the teachers to try different strategies for solving a problem allows them to focus more on their thinking processes. Statements such as "Draw the following design in two different ways..." or "Find two different strategies for ..." challenge teachers to look at problems in more than one way.

In addition to Logo, the computer section of the program uses the Geometric Supposer and other software tools for investigating non-routine problems. A problem given last year for the Geometric Supposer led one of the teachers to a rather sophisticated investigation of a particular conjecture about the line determined by connecting the midpoints of the diagonals of a quadrilateral. She later completed a formal proof and wrote a paper for publication on the process she used. Another favorite problem was the investigation of the graph of `f(x) = sin(1/x)` using CompuCalc.

Without experiencing the role of student, teachers deprive themselves of the true excitement of learning and discovery. Being enthusiastic about the subject you teach can't help but have a positive effect on your students' attitudes toward learning in general and mathematics in particular. The SummerMath for Teachers program succeeds in encouraging teachers not only to develop new understandings of mathematics but of learning itself.

References

Papert, S. "Teaching Children To Be Mathematicians Versus Teaching About Mathematics." *The Journal of Mathematical Education in Science and Technology 3* (1972): 249-262.

Simon, M.A. "Philosophy of SummerMath for Teachers." Working paper, Mount Holyoke College (1986).

Simon, M. A. "The Teacher's Role in Increasing Student Understanding of Mathematics." *Educational Leadership* (April 1986).

Playing with Language Through Logo Projects

Alison Birch
Terrapin, Inc.
222 Third Street
Cambridge, MA 02142
© 1987 Alison Birch

Logo is frequently described as having two parts—turtle graphics and list processing—possibly because the two *parts* are usually approached in entirely different ways. Although the Logo philosophy of "no threshhold, no ceiling" does, in fact, refer to both aspects of Logo, turtle graphics is often taught with too low a ceiling, and list processing is often taught with too high a threshhold.

Usually we learn Logo through turtle graphics, which is known for encouraging playful exploration. In turtle graphics, students are encouraged to experiment with and explore the turtle's environment. However, this same playful approach is rarely used when learning Logo's words and lists. People who have learned about words and lists—Logo's list processing capabilities—are often thought of as Logo experts. But, is it okay to be a *beginner* in words and lists? The answer is yes. People should be encouraged to explore this new environment with the same playfulness they experienced when moving the turtle around the graphics screen.

Through creative projects and activities, you and your students can have fun *playing* with language. The activities you use to get kids (and yourself) started in the world of "words and lists" can be quite simple. The key is to use these simple examples as means for creating learning opportunities for your students. This presentation will provide many ideas for interesting projects from my recent book, *The Logo Project Book: Exploring Words and Lists*, that will encourage interest in language and an enjoyment of words.

Instead of moving the Logo turtle, we can start by using the primitive CURSOR to move the Logo cursor on the textscreen.

```
TO COMPASS
CURSOR 17 0 PRINT "NORTH
CURSOR 0 10 PRINT "WEST
CURSOR 35 10 PRINT "EAST
CURSOR 17 20 PRINT "SOUTH
END
```

Using a simple tail-recursive procedure and combining CURSOR with RANDOM can give some surprising resu!ts!

```
TO BUSY
CURSOR (RANDOM 37) (RANDOM 23)
PRINT "BUSY
BUSY
END
```

What words would be appropriate to print in place of the word BUSY? Try printing the word SLOW instead. Does it seem strange?

Your students can practice some rules of grammar when they learn to take words apart and put words together. For example, write a Logo instruction that will take the word SMART and make it SMARTER or the word TOUGH and make it TOUGHER.

```
?PRINT WORD "SMART "ER
SMARTER
```

Write a procedure that takes any word (what kind of words do they have to be?) as input and adds "er" to the end. You can call it BOAST or BRAG or even SUPER (for superlative).

```
TO BRAG :ADJECTIVE
PRINT WORD :ADJECTIVE "ER
END
```

Where you go from here really depends on you and your students. The procedure BRAG, of course, won't work for all words. What about a word that ends in "y" for example? Discuss this with your students. You may want to write a procedure just for words that end in "y."

```
TO BOAST :YWORD
PRINT WORD BUTLAST :YWORD "IER
END
```

Or, you may want to write a superprocedure that first tests the word and then calls either BOAST or BRAG.

```
TO SUPER :ADJECTIVE
IF (LAST :ADJECTIVE) = "Y BOAST :ADJECTIVE ELSE BRAG :ADJECTIVE
END
```

Eventually, your students may even want to create their own superlatives, as in

```
I AM FUNNY
YOU ARE FUNNIER
SHE IS FUNNIEST
THEY ARE THE FUNNIERIEST
AND WE ARE EVEN MORE FUNNY THAN THAT!
```

For some projects, you may need to supply your students with several tool procedures. A few useful ones to have are PICK (which picks a random item from a word or list), REVERSE (which reverses the order of a word or list), SHUFFLE (which rearranges the elements in a word or list) and REMOVE (which removes an element from a word or list). Younger children can use these tool procedures as if they were primitives; older students may want to try writing these procedures on their own. If so, you will want to have some subtasks in recursion for these students to try first, or give them the procedures and have them think through the definitions.

These tool procedures open the door for trying some new ideas—for example, anagrams. Write a procedure that will rearrange the letters in all your students' names, then have the students unscramble them. To make this more interesting, have each student print out a list so that everyone has a different list.

```
TO SCRAMBLE :CLASSLIST
IF EMPTY? :CLASSLIST STOP
PRINT SHUFFLE FIRST :CLASSLIST
SCRAMBLE BUTFIRST :CLASSLIST
END
```

Using SCRAMBLE as a way of reviewing new vocabulary words can renew students' enthusiasm.

Students are intrigued by anything that uses their own name. Just take a program like TRAITS, for example, to get students immediately interested in using words. The TRAITS program will accept any name as input and take it apart letter by letter, matching each letter with an associated word from a list of adjectives.

```
TRAITS "FRED

F IS FOR FUBSY
R IS FOR RAPACIOUS
E IS FOR ELOQUENT
D IS FOR DROLL

THESE ARE YOUR TRAITS, FRED.
```

Another example of a fun project is called "Make a Monster." This project uses the idea of generating random sentences and extends it to generating random descriptions of strange creatures. Using the tool procedure PICK, words are picked randomly from lists and put together to form descriptions.

```
A SHORT GREEN SCALY TREE TOAD
A LARGE PURPLE FUZZY CAVE BIRD
```

Young students will be excited to change the words, have the computer print a description and then draw their own interpretation. Older students may want to extend the program to have it print more involved descriptions.

```
A TALL ORANGE SLIMY JUNGLE CREATURE WITH 5 ARMS AND FANGS
```

On a more serious note, your students may want to simulate a debate between political candidates by picking randomly from lists of phrases to create authentic sounding sentences. The goal is not to teach Logo programming, but to use Logo as a tool for letting kids become actively involved in learning and excited about language.

Learning about Logo lists leads to all sorts of ideas—writing a program to act as a dictionary or thesaurus, secret code programs, a memory testing program, investigating probability using lists of cards, database programs, fortunes and horoscopes, quiz games and so on. The number of ideas is endless. As you learn more about Logo's list processing features, you can both expand on earlier projects and think of new ones. Logo provides an environment that encourages play and exploration in all areas and at all levels.

A Book with 50,000 Pictures: Logo and Videodiscs

Glen L. Bull and Paula S. Cochran
Curry School of Education
University of Virginia

UVA/IBM Institute

We began to explore the possibilities of interactive videodisc technology and Logo during the UVA/IBM Institute on Learner-Based Tools for Special Populations last summer. One of the goals of the Institute was to help working teachers and clinicians develop open-ended, flexible applications for computers. By combining features like graphics, synthesized speech, and switches, teachers can use Logo to tailor a computer environment to the particular needs of a student or class. The UVA Logo Videodisc Tools seem to be one of the most promising applications resulting from the Institute, and Institute participants continue to field test applications in their schools and clinics.

A Picture is Worth 1000 Words

The acquisition of language has a benefit which most of us take for granted. Language allows us to consider and discuss objects, people, and situations not only in past tense, but also as we plan for them to be. Language permits us to manipulate ideas and think about things that are not limited to our current time and place.

To do this with language, we are required to recognize and use symbols of some sort. As we learn language, we learn to use an abstract representation for a concrete object or event. For example, many children and adults have learned to use the word FORWARD to cause a spot of light on the computer screen to move a specified distance.

In teaching, often the goal is to to help students move from a concrete representation of a concept to a more abstract representation. The most concrete representation would be the real object or event itself. An abstract representation which is frequently the goal in the classroom is a written word. (In a math or science class, the goal may be the derivation of a principle or formula which is an even more abstract representation of the event). Different teaching materials and stimulus items engage different senses (do you feel it, see it, smell it, or hear it?). They can be arranged in a hierarchy from concrete to abstract, such as this:

CONCRETE ----------> ABSTRACT

Objects	Pictures	Words
real (truck in street)	photos	spoken
toy (truck in box)	drawings	written

Many teachers have wished for easily accessible "moving pictures" with which to illustrate and discuss some concepts. We think that a videodisc player combined with a computer and Logo can be used to address this need for easy access to teaching materials which help bridge the gap between the concrete and the abstract.

Pictures, Pictures Everywhere!

It is possible to control a videodisc player (See Note 1) with few Logo tool procedures. Now it can be as straightforward to show a real photograph on the computer screen as it was before to access a Logo graphic. It is fascinating to see real slides and video sequences incorporated into Logo programs.

An optical disc looks a lot like a phonograph record. Instead of just sound, however, an optical disc also stores frames of pictures. A videodisc player "reads" the disc and sends the signal to a monitor -- a T.V., or a composite computer monitor.

An incredible number of frames -- about 50,000 -- can be stored on a single optical disc. They can be in the form of single pictures, like slides, or videotaped sequences. It is hard to conceive of just how much visual information one disk can hold. Imagine a teacher trying to hold 50 photographs in her hands. Now imagine her holding a 500-page picture book. Now imagine there are 100 teachers, each holding a 500-page picture book. NOW, imagine yourself holding the equivalent in one hand: a single videodisc.

Logo In the Picture

Easy access to large numbers of real photos is a welcome addition to the traditional print and computer graphics materials available to teachers. For example, video-taped action sequences can be routinely used instead of still- to illustrate the difference between "The boy is hitting the ball" and "The boy hit the ball," or between "climbing up" and "climbing down."

It turns out that Logo can be used as a convenient interface for controlling a videodisc player with a computer. Why is it desirable to control it with a computer (rather than the usual remote control)? For some activities, computer control may not be necessary or desirable. However, for many educational and clinical applications it is desirable to have access to individual picture frames by name and to incorporate other features such as text and synthetic speech.

The frames on an optical disk are numbered. The number for the desired frame can be keyed in on a remote-control, and the correct picture will appear. However, remembering the frame number is not necessary when the videodisc player is used with a computer and Logo. Using Logo, it is possible to "name" the individual frames you want to use in a lesson. The Logo videodisc tools we have developed let the user see the desired frame by typing SHOW and the name of the frame:

```
?SHOW PENGUIN    (slide of penguin appears)
?SHOW PUFFIN     (slide of puffin appears)
```

The UVA Logo Videodisc Tools procedures can be incorporated into existing Logo programs. For example, a teacher could use the procedures which include "named" P pictures in the morning as stimulus pictures in reading and in the afternoon a fourth grade student could use them to illustrate Logo poems about birds.

Application to Content Areas

The UVA Logo Videodisc Tools we are developing make several other commands available for control of the videodisc player. For example, PLAY and HALT will stop and start video sequences. Imagine a lesson in which the teacher shows the student a drawing of a fish. The teacher says, "the fish is swimming." Now imagine how much more concrete the same lesson could become. The teacher and child look at the computer screen. The teacher types SWIM. A fish appears and swims around near the floor of the ocean. The teacher says, "the fish is swimming."

Access to the slides and video sequences available on videodiscs can provide teachers with "illustrations" which are engaging and informative, and easily incorporated into their previous lesson plans. In addition, however, some videodiscs may give students access and control over visual information which may make new kinds of assignments and projects possible.

For example, there is an optical disc which includes a video biography of Vincent Van Gogh, and slides of most of his paintings (Vincent Van Gogh, 3M Company). This disc has obvious utility for art teachers. However, it could also be used by students in English class to illustrate Logo poetry or short essays about a variety of topics. In French class, selected paintings could be arranged into a slide show by students, who could use Logo to include French captions.

Synthesized speech is easily incorporated into Logo/videodisc programs (see "SpecialTalk", Logo Exchange, January, 1987 about making Logo talk). We have made a Logo program we call Alpha-birds in which pressing any letter key on the keyboard causes a different picture of a bird to appear. When the bird appears, the speech synthesizer tells the child what it is. For example, when you press T, a color slide of a toucan may appear on the monitor, and you would hear "toucan".

Alpha-birds is just the beginning (inspired by the fact that we have a disk with lots of birds on it (BIO-SCI, Videodiscovery). The applications for regular and special education are numerous. For example, a physically handicapped child using Logo to access to a videodisc player may have the equivalent of personal control over a picture book for the first time. Using a switch or other alternative access to the computer, he can "browse" through a disk using the Logo tools, or make use of the pictures as part of a Logo project.

Videodisc technology is already being used by some educators to develop and refine computer-assisted-instruction programs (Hofmeister & Thorkildsen, 1987). This technology does not have to be restricted to uses which are pre-programmed. Using Logo, it is possible to provide tools which make computer control of videodiscs accessible for teachers with minimal programming skills, or students themselves.

Note 1: Thus far we have used IBM Logo to interface an IBM PCjr or an IBM PC/XT with a Philips VP 935 and a Pioneer LDV-6000 videodisc player. We have used Apple Logo II with an Apple IIe and the Philips VP 935. We have not been concerned yet with overlaying the computer and the video images, primarily because of the cost of the necessary hardware. We just set up a second, generic monitor to show the images from the videodisc player beside the computer screen.

References

Cochran, P., & Bull, G. (in press). Special Talk: Logo, The Binding for a Book with 50,000 Pictures. Logo Exhange.

Hofmeister, A., & Thorkildsen, R. (1987). Interactive videodisc and exceptional individuals. In J. Lindsey (Ed.), Computers and exceptional individuals (pp. 189-205). Columbus, OH: Merrill.

BIO-SCI and Vincent Van Gogh videodiscs available from Videodiscovery, P.O. Box 85878, Seattle, Washington, 98145-1878. Phone: 206-547-7981

THE LOGO ALGEBRA PROJECT

Ricky Carter
Lesley College. 29 Everett St. Cambridge, MA 02238

Logo Algebra is a two year National Science Foundation research and development project. Our goal is to see if, with the support of Logo and the Computer, we can make some basic ideas of algebra more accessible to students, perhaps as early as the sixth grade. One might ask, why sixth graders?

First, we believe that many fundamental ideas of algebra such as the idea of an equation and of variables are not cognitively beyond sixth graders. Second, we believe that many of the problems experienced by older students learning algebra have to do with how it is presented to them, the techniques that have been available for teaching it. Third, we believe that by using a new set of techniques we can make many of the ideas of first year algebra accessible to younger students.

If we can succeed, and our evaluation work thus far suggests that we are suceeding, it might suggest that these ideas and approaches will be useful for teaching older students as well.

We are in the final year of the project. In the first year we developed materials for 3 modules and did some pre-piloting work in 5 local classrooms. This year we are continueing development work and doing a pilot test of the materials with 2 classes of 6^{th} graders.

OUR APPROACH:

I think the driving idea behind our work can be summed up in Papert's term "appropriable". In Mindstorms Papert says that for new ideas to be learned they must be "appropriable", and for ideas to be appropriable they must be presented in interesting and meaningful contexts. And that, in a very simplified way, is our critique of the teaching of standard algebra; too many students experience algebra as a process of shoving around a set of abstract symbols according to what often seem to be a set of arbitrary rules.

For a task to be meaningful you must be able to connect it to something. The problem with X's and Y's and the rules for manipulating them in traditional Algebra is all too often there is no meaningful context to connect them to. Our first step has been to try to provide intermediary representations for algebraic ideas, to begin with pictorial or iconic images that can be manipulated on a computer screen. (This is why we have chosen the Macintosh as our development machine.)

For example, instead of beginning with X and Y to stand for unknown numbers we have used the idea of a bag of marbles (introduced in the context of a number guessing game). Algebraic notation uses X to stand for an unknown number, we begin with a picture of a bag of marbles with an unknown number of marbles in it.

For the idea of an equation and doing manipulations on an equation we will be using a balance beam as a pictorial representation for equality. If we put each side of the equation on one side of the balance beam when an equation is true the balance will balance. If you do a manipulation that does not change the equality (like subtracting the same number

from both sides) the balance stays balanced, if you do something that destroys the equality the balance tips.

Instead of looking at functions in terms of the abstract symbolism of F(x), we introduce the idea with an image of a machine that takes and input, does something to it, and produces an output. We use Logo to allow students to actually construct the "machines" by writing procedures and to put them together to model what, at a formal level, is called the composition of functions.

The idea is that once students get comfortable representing mathematical ideas and operations with these icons then standard notation can be introduced as a short hand way of expressing them.

A second part of activities being meaningful has to do with students experienceing themselves as **taking an active and generating role** in their learning. One way we are trying to give students a chance to take an active role is to have them, whenever possible, make up problems for each other to solve. Whether its creating functions in Logo for a guess my rule game, inventing a code and using Logo to write procedrues to encode messages for others to crack, or trying to create number sequences noone else can figure out students seem much more involved when they are trying to create problems for other kids or solving problems that other kids have invented.

Finally, a third aspect of our approach has been to try to tie the work in Algebra to things that will be interesting to the students, such as writing Logo procedures to produce Gossip, creating number puzzles, writing and breaking Codes, or Making Quizzes for younger students. We have tried to choose activitites that, while having algebraic content, will also have **intrinsic interest to students.**

THE CURRICULUM:
We have developed and trial taught three modules:

MODULE 1:
Module one is formally titled Equations and Unknowns, but we refer to it as the Marble bag module. Here we will introduce the marble bag icons in the context of doing a number trick - where you think of a number, I ask you to do a few calculations e.g. add 2, double the result, add 1, double that, add your original number, subtract 7. When you then tell me your final result, I very quickly will tell you your original number. We then show the students how to use pictures of marble bags and marbles to represent the story and solve it.

To find the original number students must do algebraic manipulations, but they do it using iconic representations instead of X's, Y's and numbers. The computer will provide a "marble bag" lab (written in Logo) where they can practice generating and solving marble bag number stories. For solving marble bag problems we will also introduce another graphic representation: the balance beam. The balance beam image will introduce a way to think about operating on marble bag problems and maintaining equality. It will be used to model a process that will later be used for solving equations.

Once students feel comfortable with these manipulations standard algebraic equation notation will be introduced as a shorthand way that lazy mathematicians use to record

such stories. The "marble bag lab" will then allow them to practice moving back and forth between the iconic and the algebraic representations.

Also introduced in this module will be a unit on signed numbers, using an icon we have been calling an anti-marble. Just as +2 and -2 cancel each other out, 2 marbles and 2 anti-marbles will cancel each other out. Students will play dice games involving "antimarbles" as a model for how operations with signed numbers work. In addition we will use a Logo turtle number line as a second model of signed number operations.

The other area that we will need to introduce in this module is the order of operations and parenthsized expressions. Parenthesized expressions will be introduced as one way to record marble bag stories. In order to practice these ideas students will be introduced to Logo's arithmetic capabilites and given challenges such as: using only 2, 3, 5 and the operations +,-, * ,/ and () can you write a Logo line that will give 25 as a result? Inspired by a program called "Expression Writer" Paul Goldenberg has written a Logo program which will generate a goal number and provide a list of "resources" (numbers and operations). The challenge is to write an arithmetic expression, using all the numbers and any of the operations, that will evaluate to the goal. The program can be set to present problems that will demand the use of parenthesized expressions as well as expressions involving variables. Students will also be challenged to come up with problems to set for each other.

MODULE 2:

In our second module we will approach algebraic equations from a very different perspective: programming directly in Logo, and we introduce our second icon: MACHINES. The name of the module is From English to Algebra because we begin with Logo's ability to manipulate language. We start with a change of pace: the idea of getting Logo to GOSSIP. We model the creation of a GOSSIP procedure on the idea of first analyzing and understanding what gossip is and then on the idea of creating machines that will implement our analysis.

We then move on to writing Logo procedures to generate text based quizzes that will ask student generated questions such as: "Who is the world's greatest Rock star?" (we were told unequivically at the time that it was Madonna.) After this comes writing procedures to generate simple arithmetic quizzes. This is followed by the introduction of arithmetic "equations" in which students generate complex equations for each other to evaluate.

Once students get adept at making "true" equations, and evaluating whether equations are "true or "false the idea of an unknown is introduced by covering up one of the numbers in a true equation with an "X". Students are then challenged to figure out the value of X in an equation they did not generate.

The next step is to write Logo procedures to generate problems using an "X" to cover up one of the numbers in an equation; to write Logo algebra quiz procedures.

Once students can write procedures that generate equations with an X, the challenge is to find ways to discover the value of X, and to make their understanding explicit by writing a quiz program that gives advice.

MODULE 3:

The title of this module is Functions: from codes to graphing equations. We begin by having the students take the role of cryptographers. They use machine imagery introduced in the last module to think about how to construct Logo procedures to decode coded messages that appear on their disks.

They write procedures to both encode and decode messages and eventually invent their own codes to challenge each other. The codes they will be using can be expressed as algebraic functions and the parallel is developed. One tool for cracking certain codes is to make and analyze a graph and through this the idea of graphing equations is introduced. We are currently working on embedding the coding activities in a dramatic story about messages from outer space.

The second part of this module focuses on computerized function games. First is a guess-my-rule type (that is you give me an input, I give you the output and you must try to figure out the function) Students try to figure out function rules and can write their own secret rules in Logo for other students to solve.

A variation on this will involve sequence finding. Problems of the 2, 7, 13... what comes next variety: again students will not only try to solve sequence problems, but will try to make up sequences to stump both other students and the computer, and will be challenged to create a smarter program by teaching the computer new sequences they come up with (again written as Logo procedures).

Graphing of functions will have been introduced as a tool for cracking codes, here it will also be a tool in the guess my rule games for identifying patterns in inputs and outputs. This module also includes a series of equation graphing environments that will include target games (a Logo linear equation version of Sharon Dugdale's green globs) and a dial-a-graph environment that will allow students to instantly vary the parameters of an equation using the macintosh mouse and see the resulting changes in a graph.

MODULE4:

In our final module, which is still in development, we plan to use the metaphors and ideas developed in the first three as tools for solving algebraic word problems. In one part of this students will use Logo to write sets of word problem calculation tools. A variety of algebraic word problems can be solved by combining a series of simple functions (e.g. conversion problems). Students will use the idea of function machines implemented in Logo as a way to think about, analyze, and solve various word problems. They will create sets of tool procedures in Logo and will then be able to assemble them from their "toolkits" to solve various types of problems.

In my talk I will go over the role Logo has taken in the project and share some specifics about the procedures and the sequence we have been using.

TITLE : TALKING WITH IBM LOGO

AUTHOR: DAVID J. CARTMELL, SENIOR PROGRAMMER, IBM CORP.

DATE : JANUARY 6, 1987

FOR PRESENTATION AT

"EAST COAST LOGO CONFERENCE '87"

COPYRIGHT 1987 IBM CORP.
ALL RIGHTS RESERVED

INTRODUCTION

Can you hear me talking as you read this paper? If you can you have a good imagination. How do you like to watch TV with no sound? It's nice for the ads, but you don't get much out of the mystery thriller. Have you ever been in the other room when the TV was on? Usually you can follow the program without seeing it. The reason you could follow the program was because you could hear it! Have you ever heard a turtle talk? Not very often!

One modality which is important for information exchange is the voice. In fact some of us are auditory learners. This implies that our dominant input mode is hearing information. (The visual learner's dominant mode is using the eyes.) It is being found in education systems that the optimum presentation capability is one that combines multiple modalities.

Logo's dominant modality for information presentation is visual using graphics and words. But take heart, because it is easy and exciting to extend it to include voice as well. In this manner it appeals to not only the visual learner, but also the auditory learner.

This paper describes a set of tools available for Logo which allows the "Turtle to Talk". The tools allow it to talk with a natural voice using digitized speech or with a synthetic voice using synthesized speech. It is also possible for the "Turtle to Hear" words using voice recognition.

Background

Most standard computer configurations do not include provision for "talking". Many include small speakers which are used to sound signals and noises at appropriate (?) times during program operation. Typically what is needed is a special card (or an external black box which is connected to the system via the serial or parallel port).

One such board manufactured by IBM is called the Voice Communications Option card for the IBM PC family of computers. This card includes the following functions: (1) digitized speech, (2) synthesized speech and (3) recognized speech. A modem is also supplied with the card. An Application Program Interface (API) is provided with the card as its support structure. The API is the interface through which programs like Logo use its functions.

The other requirements include a speaker to broadcast the turtle talk and a microphone for recording (turtle talk).

Logo Interface

A set of Logo tools was developed which use the API to give the turtle a voice. The interface includes three provisions. First is a way to specify a function and its parameters. This is done using a Logo procedure interface. Second is a work or buffer area with which to communicate with the API. This is done by reserving a block of storage with Logo when it is initialized. Parameters are passed back and forth from Logo to the API using this block of storage. Third is a direct activation of the API routines. This is done using a small machine language subroutine which is called by Logo.

The GOOD NEWS is that a user only has to worry about requirement 1 to use voice in his/her program. That is what makes a tool so neat! The user merely uses a procedure called "SAY" and follows it by a list of words. The turtle talks!

SYNTHESIZED SPEECH

Definition

Synthesized speech or synthetic speech is just what the name implies. It is speech which is artificially produced. Any text data stream, e.g. consisting of characters, words, punctuation marks, numbers, etc., can be voiced. Its quality is usually characterized as robotic like. A series of rules and algorithms are used to determine the voicing. The more robust the rules and algorithms, the more natural sounding and the more accurate the speech reproduction.

A First Example

(A poem with random word selections.)

Features...Considerations...Limitations

The paramount feature of 'SYNTHESIZED' speech is the capability to voice random text. A text data stream typically contains much more than words, i.e. character strings separated by blanks. It also contains numbers, punctuation, capital letters, abbreviations, acronyms, columns of words and numbers as well as other constructs. All of these characteristics must be handled by the synthesizer. The paramount requirement for synthesized speech is the production of intelligible speech. Voice quality as well as accurate pronunciation are important. Typically the voice quality sounds more robotic than natural or human like.

LOGO Interface

The Logo interface includes two provisions. First is the ability to voice text streams. This is provided by a procedure called SAY <list> where <list> contains any valid

list of characters, words, punctuation, etc. The second provision is to provide access to the various parameters which control the synthesizer. This provision is provided via a series of procedures, e.g. a procedure to change the volume: VOLUME 150

A Second Example
("Instant Logo featuring synthesized speech commands.)

DIGITIZED SPEECH

Definition

'DIGITIZED' speech is best described by analogy with a tape recorder. What you record is what you hear! You talk into a microphone for input. You listen to the output via a speaker or head phones. Instead of recording onto tape, the computer records the sounds onto a disk.

A First Example

(Recitation of a poem...both prerecorded and live during presentation.)

Features...Considerations...Limitations

As you just heard, 'DIGITIZED' speech provides a rather natural sounding reproduction of the voice. If you have a deep voice, you hear played back a deep voice...if you have a female voice, you hear played back a female voice...if you have a child's voice, you hear played back a child's voice. So naturalness is one characteristic of 'DIGITIZED' speech.

The other two primary characteristics are the quality or fidelity of the recording and the amount of buffering or storage required. You trade off the fidelity against the storage required to hold the phrases to be recorded. Three recording levels are possible with the VCO which result in three levels of fidelity, (e.g. 3600, 2400, or 1800 bytes per second).

LOGO Interface

The Logo interface for 'DIGITIZED' speech includes a series of tools to both record and playback back speech. For speech buffering, two 32000 byte storage areas are reserved. Speech is recorded in segments or phrases. A phrase is some number of seconds of recorded sound. The buffers are effectively divided into time slots. When recording, you specify the time slot, the length of the recording in seconds, the recording rate, and the name of the recording. When you playback a recording, you simply supply the name of the phrase. To playout a phrase, you use the procedure: SAY.BYNAME <name> where name is the name of the phrase.

Other LOGO Examples

(Weather forecasts with Logo featuring digitized reports.)

RECOGNIZED SPEECH

Definition

Recognized speech or speech recognition provides for controlling the computer with the voice. In voice recognition, voice is the input, and an action is the output or result.

Recognized speech involves a four step process. First a set of words or vocabulary is chosen which is to be recognized for a particular application, e.g. a set of Logo commands. Second, for each word, a character string is defined. The string can be a command or a function which causes some action to be taken, e.g. if FORWARD was the word, the character string would be 'FORWARD' which would result in the turtle being moved. Thirdly, the user performs a process to train the machine to recognize his/er particular voice saying the vocabulary. Finally, the user utters a word from the vocabulary and the system takes the action after recognizing the word.

A First Example

(Drawing a simple figure using LOGO commands recognized by the computer, e.g. a square.)

Features...Considerations...Limitations

In the English language, many words sound similar and are difficult to distinguish by recognition algorithms. There are a variety of measures which are taken to help in recognition. For example, the word set used can be so selected to eliminate similarly sounding words. Or compound words or word strings can be selected rather than simple words, e.g. words for letters when the alphabet is to be recognized. Additionally, word ordering or juxtaposition can be used. Thus rules or a grammar can be defined such that only certain words are valid at certain times. For example, when FORWARD is recognized, only a number can follow it.

A process is used to train the system to recognize the words spoken by a particular person. During the training process, each word is said in the vocabulary a number of times. The training program makes a model of the word from these utterances.

LOGO Interface

In the case of recognized speech, there is no direct interface into the LOGO program. A program is available called the IBM Voice-Activated Keyboard Utility for the VCO

card which performs the recognition function. The direct interface comes via the vocabulary of LOGO commands or procedure names which are to be recognized, their grammar and their associated keystrokes.

Other LOGO Examples

(An example of a set of words from Logo which can be recognized with their grammar and keystrokes. An example of a Logo program which can be initiated using voice recognition which triggers the use of synthesized speech showing the combing of the two types of speech.)

CONCLUSION

This paper presents several techniques for empowering Logo with voice capability. It describes tools which provide simple interfaces for the user of Logo or the Logo procedure writer. Depending on needs, the turtle uses synthetic speech to say most anything. Or the turtle speaks with a human sounding voice using digitized speech. In this case the writer or user has to record the voice tracks ahead of time. In the last example using voice recognition, a person can use Logo by speaking its commands instead of or in addition to using the keyboard.

It goes without saying, that by empowering Logo with voice, the Logo user has been empowered with voice. Now the teacher can not only add voice to his/er procedures, but also the students can explore with voice. Or the therapist can use voice in his/er procedures and the client can experiment directly with voice.

The turtle talks in several dialects. The boundlessness of the Logo language is again demonstrated. A new modality is available for exploration!

LOGO INTEGRATION

INTO AN ENTIRE SCHOOL SYSTEM

David E. Chesebrough
Sewickley, Academy

The Logo "revolution" has been successful enough to progress well beyond a few inspired, imaginative teachers who held true to the spirit of a "pure" Logo environment of exploration and minimal (but timely and skillful) teacher guidance. Many of these first enthusiasts were skilled educators in control of their own classrooms and able to create wonderful educational opportunities using Logo.

In many ways the situation has changed dramatically and the new (and old) dilemmas facing the further successful growth of Logo usage need to be recognized. Two key challenges which need to be addressed increasingly are:

> How do you successfully expand the use of Logo from the classes of one or two enthusiastic teachers to a program touching many grades and teachers?
>
> How do you support the insufficiently prepared teacher who lacks optimal motivation and who has been thrust into using Logo through a school mandate?

ONE SCHOOL'S APPROACH AND EXPERIENCES

At Sewickley Academy, a private N-12 school with around 600 students, we are making progress on both fronts. The development, successes, and frustrations of our program are worth sharing with others who are trying to integrate Logo in a thorough fashion in their school.

Director of Curriculum, Dr. Finlay McQuade led a 1982 administrative decision to introduce Logo to Lower School students. A few Texas Instrument computers with TI Logo were placed in the halls and classrooms. They were enjoyed and explored by a group of students (primarily boys), but seen by most teachers as a nuisance and intrusion. The best that could be said for them was that they were a useful learning station to some students. Without direct guidance, and diluted by being spread around, it was impossible to build any momentum in moving Logo into broader use by students and acceptance by the teachers.

In 1983 I was given a part-time position as computer coordinator and we moved the computers (now a few Apple II+ Computers) into a makeshift lab in part of the library. It was decided that all 3 - 6 grade classes were to receive exposure to Logo, and that the primary experience would be directed by the individual teachers. I took a crash course in Logo, and we hired two excellent Logo teachers for an initial inservice session. We

began with the consultants conducting several days of inservice for the teachers, and then continued with my leading weekly inservice sessions for several months.

As the teachers displayed some initial confidence with Logo, their students were brought into the computer lab to start the student program. I more or less team taught with each teacher to help develop their skills in conducting a class using Logo, and to alleviate the nervousness most of them were still displaying.

Once the weekly, 40 minute classes were begun, we waited for the promised "Logo magic" to take place. Indeed some students took off with exploring in Logo, while some others seemed to just muddle around. Also disappointing was the stagnation which began to develop with even the most interested students, as well as the inability of the teachers to become comfortable with Logo.

Recognition of the problems with our program became more and more evident during the first year. Insufficient numbers of computers, lack of a deep understanding of the computer and Logo on the part of the teachers and students, inadequate structure for the direction of the Logo program for the teachers, and the inability to keep the students excited and progressing with Logo skills all seemed to be reasons to reassess and revise our approach.

ATTACKING THE PROBLEMS

We developed several strategies to counteract the difficulties we were encountering and to improve the quality of the students' Logo experiences. We had a fundamental faith in the merit of an extended experience in using Logo and still wanted to make it a foundation for an expansive school computer program.

The following components of our present program are a result of the strategies we adopted, and could serve as a guide to other schools to varying degrees:

* I developed activity sheets which offer experiences in new Logo skills (such as introducing procedures, technical skills (printing procedures to the printer), references (color/number match-ups), exploring or project ideas, challenge shapes, or integration "hooks" (using a Logo graph on the screen to create a class related graphing exercise). The sheets are used as SUPPORTIVE RESOURCES rather than a complete substitute for the student directed, teacher supported, environment. (The materials are available as noted at the end of this paper).

* We identified a logical sequence of skill development which offer the teachers an overall structure to follow. The activities are loosely organized around levels which support an approach I call SKILL, EXPLORE, PROJECT. It has been effective to give the students new SKILLS, let them EXPLORE

with the new skills as they choose, and then encourage them to tackle a PROJECT which gets them excited and encourages them to push their understanding and thinking to new limits. When stagnation starts to set in for a student or class, we introduce them to new skills and start the cycle again.

* We sought school support for additional computers, and now have a lab which normally allows a one-to-one ratio between students and computers when desired. Even more encouragingly, teachers have come to us asking for computers to have in their classrooms for greater student access. This teacher initiated program has resulted in at least one computer (and sometimes a printer) being shared between every two classrooms.

* We developed an aggressive program of encouraging home ownership of computers, and worked out special pricing on Apple computers and peripherals with a local dealer. Many of our classes have over 50 percent home ownership of computers. In the tenth grade, where we are beginning to require computer usage across the curriculum, we have almost complete home ownership of computers.

* We have sponsored several parent Logo nights to explain our use of Logo in the school, provide hands on experiences for the parents, and sponsor a bulk purchase discount for the parents purchasing the computers. The last session resulted in a standing room only crowd and almost 40 copies of Terrapin Logo being purchased for home use.

* We are beginning to offer courses for parents on Logo so that they can have a deeper understanding of the language and the approach to using it which we encourage. Summer courses for students are being discussed.

RESULTS

MATERIALS: We have found that as an anxious teacher progresses with their class using the supportive materials, the feeling of accomplishment breeds more comfort with Logo. This reduction of the teachers anxiety translates into a more relaxed, exploratory environment - the type which Logo veterans find at the heart of the offering. In turn, the positive student response to the improved environment encourages the teacher to allow even more flexible and freer Logo use, which our materials are designed to support as well.

COMPUTER ACCESS: We now have enabled students to use Logo weekly in their class session, freely before, during, and after school in the computer lab and in their classroom, and to take their disk home to explore using a compatible computer and version of Logo.

FOUNDATION OF COMPUTER UNDERSTANDING: We now have students who are very comfortable with the computer, Logo, and basic concepts such as data disks, saving, and printing. This foundation is being used as we extend the use of Logo into higher grades and as we apply the computer to other uses throughout our program.

PRESENT PROGRAM

The strengths of what we have established have resulted in the following uses of Logo in our school:

* A learning station using the old Texas Instrument computers is available to students in the Nursery and Kindergarten classes.

* Grades 3 - 6 still have their weekly sessions, but we are starting to tap Logo for projects in classes such as math, english, and social studies. In some projects Logo is more a computing language than an exploratory environment, but we consider that a legitimate use.

* Middle school science classes in grades 7 and 8 use Logo periodically for projects (exploring bridge design and modelling, for example), or as a programming language for applications such as temperature conversions.

* Logo has been used in the senior school programming classes as an introduction to the ideas of structured programming and modularity.

* Senior school math and science teachers are examining Logo as a possible tool for exploring concepts in their classes.

PROBLEMS STILL EXISTING

I don't want to paint too rosy of a picture and give the impression that our program is right were we want it, for that is far from the truth. We are still dealing with some of the same fundamental issues, but in some cases, at a higher level.

As the computer is proving to be a productive tool in the educational process throughout our school, there are increasing pressures to use the computer for other uses beyond Logo, especially at the upper elementary grades. Word processing and the supportive typing instruction are meeting with success in grades 5 to 7. The growth in time devoted to them is pushing hard against some of the same computer access (and student interest) as Logo. Outstanding computer software, such as that with the Voyage of the Mimi series from Holt, Rinehart is doing the same.

Teachers are just now becoming creative and aware of how Logo and an exploratory environment using it can be applied to topics in the curriculum. Students still have to be made aware of the

resources we now have and encouraged to explore more on their own, rather than being spoon fed with ideas as we unfortunately discipline them to do in much of the educational process.

Another problem is that as student skills increase, the teachers are left behind again and inservice needs seem to be never-ending. This is even more of a problem as we start to have students going to the senior school with a basic knowledge of Logo. The senior school math and science teachers have little or no experience with Logo, and have a difficult time in seeing how it is applied, and/or have preconceived prejudices against it as compared to BASIC or Pascal, which they know. An exploratory Logo environment in their classes is antithetical to their whole approach to teaching and runs afoul of basic differences in educational philosophy between the lower grades and the pressured upper grades in a college preparatory private school.

In fact the interest of the Senior School math teachers offers an interesting Catch-22 situation. They want indepth programming abilities for all their students before exploring the possibilities of using Logo in their courses. However, we cannot devote that kind of time in training the students until we have some assurances from them (and class experiences) that the skills will be used extensively. A compromise may be to have a computer prerequisite to some of the higher level math classes where Logo would be used as a programming language to explore the fundamental math concepts.

FUTURE STEPS

We plan to see how far we can take the productive use of Logo as the skills of the students increase and are carried up from the elementary grades. We also hope to broaden the use of Logo with students and see if the exploratory environment which we encourage can be applied beyond the separate Logo periods. The results will be shared through future papers and conference presentations.

ADDITIONAL RESOURCES

I would be glad to share details on the program with interested individuals or schools. Write to David Chesebrough, Computer Coordinator, Sewickley Academy, Academy Ave., Sewickley, PA 15143.

The materials which were developed to support our program have been published by J. Weston Walch, Publisher under the title of THE LOGO SUCCESS KIT. The format is a reproducible teacher sourcebook with background information, resources, and activities coupled with a teacher/student resource disk for either Terrapin or Apple Logo. Ordering information can be obtained from J. Weston Walch, Publisher, 321 Valley St., P.O. Box 658, Portland, Maine 04104-0658, or by calling 1-800-341-6094.

The Mathematical Turtle: Logo in the 4-8 Math Curriculum

Sheila Cory
Coordinator of Educational Computing
Chapel Hill-Carrboro City Schools
Chapel Hill, North Carolina 27514

In 1983, the Chapel Hill-Carrboro City Schools district, like so many other school districts, was at the initial stage of using computers in the classroom. With an average of two computers per school, untrained faculty, no computer-related curriculum, and little software, it was obvious that much work needed to be done. In the few intervening years, the school district has been successful in involving all teachers at certain grade levels in using the computer for instruction through a program of intensive curriculum and staff development. This presentation will focus on the Logo component of our computer-using curriculum, giving attention to the factors of curriculum development, staff development, management, and evaluation.

The original impetus for a Logo project came from a 1983-1984 school district committee which developed a plan for computer use for the K-12 school district. This committee felt that all students should be taught to (1) recognize what programming is, (2) understand a given or self-devised problem, (3) write an algorithm to solve the problem, and (4) translate an algorithm into a computer language and use, analyze and debug the resulting program. It was suggested that the approach to programming would involve tracing, modifying, creating, and debugging programs. It was decided by this committee that programming instruction should begin at the elementary school level, and that Logo would be an appropriate computer language for students to use. The committee suggested that the responsibility for instruction be given to the classroom teacher, recognizing that a high quality of staff development and curriculum development would be crucial to a successful implementation.

Utililizing grant-generated funds, curriculum materials were written during the summer of 1984. These materials took the form of six workbooks targeted to students in the intermediate grades. They were designed to introduce students to Logo graphic commands The teachers involved in the development of these workbooks tried, through the design of the materials, to minimize two implementation problems. First, because the materials would be used by teachers who were new to Logo themselves, they were designed to be easy to teach with. The teachers developing the materials worked to resolve the dilemma of how much should legitimately be sequenced by the teacher and materials, and how much learning should be sequenced by the students themselves. The second implementation problem concerned how to fit Logo into an already-crowded school day. To address this issue, the Logo lessons were designed to be as mathematical as possible, and time for Logo was taken, to a great extent, from math instructional time.

During the 1984-85 school year, all fifth grade teachers were given released time for staff development in Logo. They were trained with the very materials that they were to use with their students. A lab of fifteen computers was established for one third of a year in each school, and student instruction began. Teachers provided feedback about additional features they would like to see in the materials. They wanted not just Logo activities, but narrative that would introduce new concepts to their students. They wanted more activities, an example of a possible solution to each activity included in the workbooks, more mathematical concepts introduced, and more teacher-support materials. To address these identified deficiencies, further development of the materials took place. These enhanced and refined materials ultimately became the student text and teacher's manual named LogoWorks: Lessons in Logo and published by Terrapin.

Beginning with the 1985-86 school year, Logo was taught to all students in grades four and five, by their classroom teacher. Logo is currently taught in computer labs, consisting of approximately fifteen Apple IIe's and a few printers. Each lab also contains a set of fifteen

LogoWorks books, labelled file disks, labelled activity disk, Logo utility disks, Logo language disks and appropriate posters. Each fourth and fifth grade classroom has fifteen LogoWorks books and a copy of the teacher's manual that goes with the book. Teachers bring their students to the lab two or three times a week for the entire school year, with each lab session lasting thirty to forty-five minutes. Teachers are also advised to use non-lab time for Logo instruction and specific techniques for doing so are discussed and demonstrated in the staff development sessions. Each computer lab has a person assigned to it as computer lab aide. In some cases, this person is a PTA-paid aide who works ten to fifteen hours per week. In other cases, this person is an existing member of the school staff. The aide is responsible for computer maintenance, scheduling the lab, initializing disks, helping students print out their work, etc. The aides are a critical component in the success of the program.

Although there is a great deal of flexibility in the Logo program, it is suggested that fourth grade teachers cover chapters one through three of LogoWorks, and that fifth grade teachers cover chapters four through seven. It is also stressed, however, that teachers should work to establish depth of understanding and not simply push forward to cover more material. If teachers adhere to the suggested schedule, fourth graders learn such things as commands to move and rotate the turtle, about angles, how to use the REPEAT command for squares, rectangles and other patterns, how to write procedures, the power of subprocedures, editing, how to manage their workspace, and how to save their procedures. Fifth graders learn much about polygons, new ways to think of circles, one approach to problem solving, and a little about variables and recursion.

An evaluation of some aspects of the computer education program for the school district was completed in June of 1986. This evaluation did include the Logo project. Because the project is still at an early stage of implementation, the focus of the evaluation was on process rather than outcomes. The purpose for the evaluation was to monitor the early implementation of the program and to gather information for further development of the program. The instrument used for the evaluation was a forty-question district-designed questionnaire administered to all of the Logo teachers. Results represented teacher perception about (1) how knowledgable they are about various aspects of the Logo language, (2) the effectiveness of Logo as a tool for teaching various concepts taught as part of the traditional curriculum, (3) general issues of system level coordination of the program, staff development, faculty support-system, and curriculum materials, (4) the responsiveness of various sub-populations of students to Logo instruction, (5) general issues relating to Logo as an instructional tool, and (6) their ability to do a good job of teaching Logo to their students. Based on the data gathered from the questionnaire, the evaluation conclusions were as follows:

- a 2-day Logo workshop is sufficient to give most teachers the skills they need to effectively address our Logo curriculum
- teachers who teach the chapters covering circles and problem-solving need additional staff development
- LogoWorks should continue to be used for our curriculum materials for this project
- Logo should continue to be part of our elementary school instructional program
- we should continue to consider moving Logo from the 4th and 5th grades to the 5th and 6th grades, but it would be premature to do it at this stage of the program's development

The information was used for decision-making about the program for the 1986-87 school year. The evaluation showed that teachers, on the whole, felt very positively about the project and would continue to work to see it improve. They felt that their skills were adequate for addressing the Logo curriculum, but were also anxious to partake of any additional Logo staff development that was planned.

As teachers became more and more involved in Logo instruction, concern arose about how their students would be involved with Logo beyond the fifth grade. To address this concern, a grant proposal was jointly submitted by the University of North Carolina Center for Math and Science and the school district. The purpose of the grant was to develop packets of materials to teach concepts in mathematics using what students already knew about Logo.

With grant funding, three different packets, each covering one area of mathematics, were developed in the spring of 1986. The areas of mathematics are (1) ratio and proportion, (2) geometry and measurement, and (3) problem solving. These broad topics were selected because teachers identified them as math topics that tend to give students difficulty. The materials were written to be used in addition to other methods of teaching these concepts. They were primarily written to address the instructional needs of students who have difficulty with these math topics. However, once they were designed, math teachers felt they would be enriching for other students as well.

These teaching and learning materials were designed to expand the impact of the Logo curriculum. In these materials, the learner objectives are math objectives. It is assumed that students have already learned the basic syntax of Logo, and that they are comfortable with writing procedures to solve problems. Each packet consists of approximately seven lessons. Each lesson contains teacher's notes, a narrative that introduces the math concept, some off-computer activities that reinforce the math concept, and some on-computer activities that give students the opportunity to apply what they know about the math concept. Although the lessons are very structured, in that a concept is specifically taught and practiced in a carefully sequenced manner, the Logo activities are relatively unstructured and there are many possible solutions to each problem.

The packets and lessons in the materials are as follows:

Geometry and Measurement
- Basic Terminology
- Naming Angles
- Supplementary Angles
- Measuring Angles
- Lines
- Quadrilaterals
- Linear Measurement

Ratio and Proportion
- Ratio and Line Segments
- Ratio and Squares
- Ratio and Regular Triangles
- Ratio and Regular Polygons
- Perimeters and Ratio
- Proportions
- Similar Polygons

Problem Solving
- Problems Using Addition
- Problems Using Subtraction
- Problems Using Multiplication
- Problems Using Division
- Deciding Which Operation to Use
- Estimation

It has been suggested by people who are both knowledgable and thoughtful about computer education that it might not be feasible to have high-quality Logo instruction unless

Logo is taught by well-trained specialists. The Chapel Hill-Carrboro City Schools are making excellent progress toward high-quality Logo instruction, using the classroom teacher instead of well-trained specialists. They are doing this through an emphasis on curriculum development and staff development. By so doing, they are also empowering all teachers to use the computer and Logo as another tool for classroom instruction in mathematics.

NO LIMITS LEGO® TC logo
John E. Crowley, TECHNIC Sales Manager

In November of 1983 a working partnership began between Professor Seymour Papert of M.I.T. and LEGO SYSTEMS, INC. Professor Papert is the author of Mindstorms Children, Computers and Powerful Ideas and the developer of the Logo computer language. Both Papert and LEGO SYSTEMS, INC. are actively interested in developing creativity and thinking skills in children. Thus, together they embarked upon a three year development program culminating in the development of LEGO® TC logo. Using LEGO® TC logo, children are able to build machines with special LEGO® elements, then write Logo computer programs to control the machines they have built. Seven diverse schools in New York City, Hartford and Boston have been actively involved in school testing over this time period to ensure that LEGO® TC logo is a learning tool of the highest quality.

The linking of the concrete to the abstract; the real world environment to technology and computers is what LEGO® TC logo is all about.

LEGO® TC logo is the newest component in an educational system developed by LEGO SYSTEMS, INC. The system can take students in younger grades from recognizing symbols, shapes and farm animals using LEGO's Early Childhood Products through computer controlled models. The TECHNIC Sets begin with Set 1030 which describes how things work such as levers, gears and pulleys. Set 1032 motorizes the components and shows students real applications for the simple machines. In fact, TECHNIC has been written into the Simple Machines Curriculum in New York State and the Problem Solving Curriculum in Minnesota.

Sets TC0, TCI and TCII add the computer interfacing to the mixture for computer controlled robotics. All three sets contain LEGO building elements. LEGO® TC logo is the special Logo programming language developed by Papert to enable the models built from these kits to move. The children then have precise computer control over their LEGO constructions. The possibilities become limitless.

LEGO® TC logo assists in learning the abstractions that the computer presents by linking it to a three dimensional object, the LEGO model. This is a new approach for traditional computer teaching. Children no longer work in isolation with the micro. Each LEGO Educational Set is designed for group activity. This group activity facilitates the acquisition of interpersonal skills and language development. This is a totally different approach from traditional drill and practice software. The students learn "project skills".

LEGO® TC logo is an interdisciplinary activity involving important ideas from mathematics, engineering, architecture and science. While working on LEGO® TC logo projects, students experience what it is like

® 1987 LEGO Group, LEGO is a registered trademark of Interlego A.G.

to actually work as a designer, inventor and engineer. They create new ideas, build prototypes, test them out and modify them to improve performance.

Using LEGO® TC logo, students learn to experiment through trial and error. When a machine is not structured properly, it will fall apart. Students discover for themselves the reason why this happens. They experience it rather than being told about it or being shown pictorially on a blackboard. Many principles that are difficult to describe and hard to understand are easily explained via three dimensional objects. Researchers have documented that learning abstract principles is best achieved through concrete, hands-on experience. LEGO® bricks and logo are similar in structure. Both systems use building blocks as foundations.

To use LEGO® TC logo, students use a special interface box to which they can send signals to LEGO motors and receive information from LEGO touch and optic sensors.

To program their machines, students use a special version of logo; using primitives such as ON, OFF, RD and SENSOR?. Papert developed LEGO® TC logo to use natural, intuitive language; thereby making programming easier and more natural for children.

Each Educational Set comes with activity cards. The cards can be used as a foundation for a lesson. The front page shows reality in a picture, whether it be gears or conveyor belts. The inside shows the building process for up to 3 models - easy to difficult. There are no written or graded instructions. Teachers can design their questions for appropriate grade levels. The back page of the activity card features a more difficult model for talented or gifted students. Quicker groups can work on the same topic as slower groups facilitating easy classroom management.

The LEGO elements, contained in each kit are very durable, made from hard ABS plastic, which is the same material that football helments are made of. Each element is machined tooled within the tolerance of a single human hair.

The LEGO® TC logo Sets are also equipped with tested Student Guides and a Teacher's Guide. Within these guides are reference pages for each building activity, suggestions for presentation of lessons, further activities, topics for discussion and sample questions to help students get their creative juices flowing.

Children have used LEGO® TC logo in a wide variety of projects. One 4th grade girl, for instance, built an automatic door which opens whenever a touch sensor is triggered. A 5th grade girl built an entire chocolate candy factory, complete with computer controlled conveyor belts. Other children have designed vehicles that automatically change direction whenever they bump into walls.

The essence of LEGO® TC logo is summed up here by Professor Seymour Papert:

"My vision of how computers could best be used to improve learning is totally different: I think the child, not the computer, should be in charge. Instead of the computer programming the child, the child should program the computer, and I have followed through on this idea by designing languages children can use to program computers. The unimaginative use of computers in education is a torment for children. However, properly harnessed, computer technology can enrich all aspects of children's lives and awareness. The edge of imagination is the only limit."

STARTER PACK CONTENTS
TECHNIC Control Set and Storage Case
Apple IIe slot card and cable*
Interface and transformer
3 different student guides (2 copies of each)
1 Teacher's Guide
1 Reference Guide
1 Quick Reference Guide
2 Diskettes

** Approx. $400

*All our peripherals are compatible with Apple IIe, the dominate classroom computer.

**All items available separately also.

For information write or call: Sarah McDonald
Educational Division
LEGO SYSTEMS, INC.
555 Taylor Road
Enfield, CT 06082
203/749-2291
800/243-4870 (Except AK,HI,CT)

Associated Learning: Logo Primitives in Real Life

Beverly Cunningham
Council Bluffs Community Schools

Lee Cunningham
Educational Computing Innovations

As we read Mindstorms we become very familiar with the term syntonic. Papert defines body, ego, and cultural syntonic. Actions are body syntonic when they can be directly related to one's own body. Ego syntonic refers to the relationship between an outside source and the individual's feelings. Cultural syntonicity occurs when the movements of the turtle can be generalized to the child's activities.

The word syntonic takes us on an interesting trail of discovery through the dictionary. It is of Greek origin. The Greek word is syntonos which means "being in harmony with". One lexical definition is possessing a temperament normally responsive and adaptive to one's social or interpersonal environment. Syn means "with", "opposed to anti", "together", "by means of", "at the same time as", or "alter". Syntonic does, indeed, describe Logo.

Logo literature is full of ways to introduce children to Logo in syntonic ways. However, when we go past the initial primitives there seems to be little written to indicate how, or if, to continue syntonically, with the exception of playing turtle.

The majority of Logoites wholeheartedly agree with Dr. Papert's ideas of children learning best through associated learning. In fact, the terms associated and syntonic are nearly synonymous. Many educators and society, as a whole, adhered to associated learning long before Logo. This is evident in the almost lost art of young men becoming apprentices, journeymen, and finally full craftmen. Years ago girls didn't have the opportunities to take courses to help them become successful homemakers. These skills were learned from their mothers when the need arose. The vast majority of our ancestors learned solely through associated learning. This style of learning should, in some words of Jean Piaget, make "equilibration", which is the constant effort towards a balance between "assimilation" and "accommodation", quite effective in building the intellectual structures that Seymour Papert speaks of in Mindstorms.

Hopefully, as more educators become familiar and work with Logo and its philosophy, we will see a trend toward reversing the current philosophy of filling students full of disassociated facts. Our children need the experiences provided in an associated learning environment if they are to meet the challenges of the future. Facts alone will not be sufficient.

Copyright 1987 Beverly & Lee Cunningham All Rights Reserved

A syntonic introduction to almost all primitives and special keys is possible and is an almost assured way of students understanding the use and gaining proficiency fairly quickly. This goes back to the idea of associated learning. We all learn best when we have the proper foundation, have a desire to learn, and can relate the new material to the environment in which we are functioning.

Playing turtle and using FORWARD, BACK, RIGHT, and LEFT are excellent examples of body syntonicity. This serves as an outstanding way of not only helping students get off to a good start but also building the foundation for more sophisticated association. As students progress past the initial phase of Logo, playing turtle must take on aspects of cultural syntonicity. Yes, body and culture do overlap, but in looking at ways to help students understand and relate to primitives and special keys, we must look at the cultural aspects of syntonicity.

On the surface, STOP rules make little sense and most students have absolutely no idea how to play turtle with them. Let s take a trip to PE class, real or imagined, and find out what a real life STOP rule is. The scenario: the class is running the 100 yard dash to determine who will compete in the all school field day. The coach sets up the parameters of the race, the starting point, the ending point, and any conditions that are to be met during the race. Most students, even fourth graders, can relate this to Logo STOP rules. The understanding begins by relating a cultural experience to the unfamiliar STOP rules. The relationship promotes an understanding and reason for the new primitive. It also makes playing turtle with STOP rules easy.

There are some children who shudder at the idea of competing in a 100 yard dash and consequently will probably not relate to it favorably. These students will appreciate a different analogy. For instance, you are walking down the sidewalk and want to cross the busy street. Consider the necessary parameters for safely crossing the street. You must cross at a crosswalk with traffic signals and you must wait for the green light (or the walking man) and you will need to hurry when you receive the proper signal. Others may be more comfortable thinking in terms of making a home run in baseball. Children who live in rural areas may have difficulty with the busy street analogy. Microwave ovens are good examples of STOP rules. Some students may need several frames of reference before they can effectively associate an experience with STOP rules.

When students can relate to primitives we lower the use threshold. Most teachers wouldn't seriously consider introducing SETPOS to the average elementary age child who has spent only thirty minutes or so with Logo. When SETPOS is introduced in a syntonic manner young children pick up the concept rather easily.

A learning disabled fifth grade girl who had worked with Logo about thirty minutes joined two talented and gifted fifth grade girls who were staying in at noon to learn about SETPOS. An outside observer would have been unable to determine which girl had very limited experience. We will share the SETPOS technique during the presentation.

You can tell by the look on a student s face when the appropriate association has been made. Computer partners will often continue to discuss the association as they work and it appears to continue to assist them in the execution of the primitive and understanding of how it relates to the rest of their procedure.

The more we, as educators, use associated learning techniques to make primitives and special keys meaningful to students the easier it becomes. We don t want to overlook the creativeness of students in helping us find ways of associating primitives with real life. They often come up with very innovative ideas of their own. A sixth grade girl was watching a group of third graders. They were having some trouble erasing an unwanted line when one of them pleaded with the turtle, "Oh, please erase." They turned to the older girl and asked how they could erase the line. Her response, "You just said Please Erase. What's the short way to write that?" More than a year later these children still jokingly refer to PE being short for Please Erase (LCSI Logos).

We all need a "bag of syntonicity" in our repertoire of Logo knowledge. Each of us is different and relevant comparisons for one may be completely off base for someone else. At times several different metaphors may be useful in helping students relate to a specific primitive or special key. Jean Piaget has indicated that every child is an inventor. Any child only differs from Archimedes, Picasso, and Einstein in degree not kind. The former may discover something new to him/herself while the latter discovers something new to us all. After forty years of intensive research in school learning, Benjamin S. Bloom stated, "What any person in the world can learn, almost all persons can learn if provided with appropriate prior and current conditions of learning."

Many of us have well organized Logo tool kits which we feel are essential to our work with Logo. We would do well to have a "syntonicity kit" as an aid in helping us to help students in Logo and other forms of computerized associated, learning. For us, as educators and parents, to provide associated learning, what we present must be syntonic or, in other words, on their wave length. We mentioned that what is syntonic for one student may be clear off base for another. For those who remember the days before cable TV or who still live out in the country, remember turning the TV to a channel and getting no picture or audio at all? You

were too far away and/or the antenna was pointing in the wrong direction or was insufficient to let the set receive the signal. What happened to the TV happens to many students. Perhaps they receive a picture and audio but neither are clear. They can get some of what is going on but it is fuzzy and confusing. This often happens to children as we educators are near but not on their frequency and therefore it is not completely syntonic and consequently not good associated learning for them. In this latter case, fine tuning will often solve the problem for the TV and so will it for student associated learning. We need to do a lot of fine turning for our students. Today most of our TV s come cable ready for a perfect picture and are connected to the cable which is precisely syntonic for them. Unfortunately most of our students don't come "cable ready" and we in the educational system don't emulate the cable as well as we could and should.

Ten Tried and True Turtle Technical Tips

Beverly Cunningham
Council Bluffs Schools

Lee Cunningham
Educational Computing Innovations

Having followed the "Logo Conference Circuit" for a few years and presenting at many of the conferences, we have enjoyed benefiting from and contributing to the open exchange of helpful information. Within Logo and beyond, we came to appreciate that many little things, some rather subtle, take a lot of our busy time and all Logoists we know are very busy people. We decided to find more useful ways of gathering and exchanging ways to avoid pitfalls that take us away from our prime Logo educational efforts, yet are things that, at times, must be done if we are to achieve our goals. This was further brought home by some of Tom Lough's comments in his Keynote Address at the 1986 Logo Institute Weekend Conference in reference to how some of the "little things count so much". They certainly do and we cannot live, or often proceed, without them. It is our hopes and desires to collect and disseminate these little things in a way that will make life easier for all our Logo friends.

Have you ever been frustrated because you couldn't control the computer? Perhaps someone told you how to do something, it sounded so easy, and now you can't remember. Or perhaps, after hours of research you came up with the solution to your problem. It worked beautifully and you knew you'd never forget the process. But you did because you failed to write down the instructions.

We suffered from these problems and a new rule was established in our home ---- if it works write it down, clearly, concisely, and step by step. We have dictionaries for seldom used words. Doesn't it make sense to have a "computerary" for computer operations we seldom use or those little things off the beaten path that are hard to find in any literature when we need them. When you write the instructions don't forget to indicate the Logo version to which they apply. Several years ago there was an outstanding article about an interesting operation but the Logo version was not mentioned. Many of us may have spent unwarranted time with this article not knowing it wasn't compatible with our version.

The majority of computer and Logo books overlook the possibility of the user wanting to utilize more than one disk drive, convert from one DOS to another, or from one Logo version to another. The list is endless and includes quickies as well as items that require rather extensive documentation. Many of these operations are generic, others apply to one or more versions of

Copyright 1987 Beverly & Lee Cunningham All Rights Reserved

Logo. The knowledge of an operation that will enhance or make your work easier is often a comforting thought as we work to provide appropriate learning experiences for students. The majority of these operations are relatively easy if the proper documentation is available.

Sections of Logo books that describe advanced and rarely used primitives had a tendency to give us a feeling of inadequacy but not any more. We teach many of these formerly difficult primitives in beginning classes for teachers and at the elementary school level. In fact, many of them are extremely useful to beginners since they are powerful ways to correct mistakes more advanced users don't often make. Examples are BURY, REMPROP, and PPS (Apple Logo).

Advanced primitives are usually not difficult to understand and use. The difficulty lies in them being labeled "advanced" or "rarely used". We believe they are so labeled simply because there is very little information available on how and why to use them and that information is often confusing or written for "advanced users".

When one takes the time to sort out the many Logo and Logo related computer operations many of Logo's frustrations are eliminated. This is far easier said than done. One of our first real exposures to this came several years ago when we wanted to put print procedures in the Startup file of the original Apple Logo. It sounded so easy, but it didn't work. Several hours and reference books later we had accomplished our goal. What a proud moment! And how discouraging when we tried the same thing a few months later. This was the beginning of our "computeracy". A three by five index card posted on the bulletin board at school gives students all the information they need to PACKAGE, BURY, and UNBURY with little if any help. This information will be shared during the presentation.

DOS is very important for us to use if we are to receive maximum benefit from Logo. There are a number of file handling utilities that are very useful but virtually unknown to many Logo users. During a sharing of Logo programs, it was suggested to a computer teacher that because of the nature of a particular program it would be best to use FID to make a copy. The teacher knew absolutely nothing about FID. This lack of knowledge could have resulted in her inability to utilize a very good Logo program. This was in a school where all outward appearances suggested they were very progressive in computer education.

Introducing simple computer tips to beginning students often gives them the opportunity to realize the power of the computer and exercise greater power over the computer. This is particularly valuable if the child is a slow learner or is hesitant about

his/her chances for success at the computer. Two monitors, one black and white and one colored, connected to the same computer can be a real plus in the classroom. The color monitor is great for Logo graphics and programs that require a color monitor. The resolution of a color monitor is often too poor for word processing. The black and white monitor can be used for word processing. The two monitors are an excellent way to show students the difference between Logo color on the two monitors. One printer will accommodate a variety of uses on one computer.

Demonstrating the proper care of disks is an ongoing activity for anyone who teaches computer classes or works with students using computers. We can tell children over and over how delicate disks are but seeing is believing, especially for younger students. Open the outer cover of a disk and let students take the shiny disk out and explore it with their fingers. They'll see that dirty hands make smudges, pencils scratch, and hair clings. Touching and feeling are great teachers. A third grader came in one morning before school, carefully and firmly took hold of the disk with his fingers on the window, and deposited his breakfast jelly! One piece of commercial software saved is worth the time to take an old data disk apart so students can see it is delicate.

There is an easy way to start programs automatically. You have a DOS 3.3 Apple Logo program and want to use it with your Terrapin Logo but don't have time for all that typing. Yes, there is a way to convert programs. You can convert any Apple Logo, Terrapin Logo, or Apple Logo II program to the Logo of choice. Original Apple Logo users, have you been frustrated with a workspace filled with buried properties (not just procedures)? There is an efficient way to remove them. We'll discuss these and other technical tips during the presentation, some quickies and other more involved. They will be presented in a step-by-step manner. Questions will be answered and you'll go home with the necessary information to use these tips immediately. This session will be useful to users of Apple Logo, Apple Logo II, Terrapin, and LogoWriter.

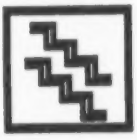
A Logo Curriculum Based on Critical Thinking and Problem Solving Using the Polya Model

by Lis D'Introno, Stephen D. Mosher,
Christopher Van Leeuwen, and George Collison
Holyoke Public Schools

The heart of our curriculum is the development of critical thinking and problem solving skills through the Polya model. Logo along with other applications such as word processors are used as tools to aid children in developing and refining their thinking. Our goal is to develop good thinking that is both critical and reflective; computer literacy or skill in programming is a byproduct, not a goal. We are using computers to make students active participants in their own education.

Using specially constructed Logo mazes, fourth grade students begin Logo instruction with a simplified four step Polya model: Define the Problem, Make a Plan, Carry out the Plan, and Review the Result. We stress the first two steps of the model, careful definition of the problem and thoughtful choice of appropriate stategies. Students are introduced to three simple problem solving heuristics: Breaking the problem into parts, Solving by finding likeness (analogy), and Investigating by sending to limit. Students monitor their progress through the model by referring to an representation of the Polya model at each computer station and on all walls in the room (see figure 1). The icons symbolize the main aspects of each of the steps in the model.

Figure 1

Step One	Step Two	Step Three	Step Four
			
Define the Problem List, Diagram, Sequence. Restate the problem clearly.	**Make a Plan** Choose a strategy: By parts, by likeness, or Investigate by limits	**Carry Out the Plan** Check each step. Use your errors to help find a solution.	**Review the Results** Verify your results. Try for a second solution using another method.

The use of Logo to develop critical thinking and problem solving skills presents a pedagogical problem. Tom Snyder and Jane Palmer are correct in describing Logo as a tool with no built in rules or pointers to intellectual domains. It is largely ineffective when children are left to their own devices - Logo as a sandbox. To bring about the developmental promise of Logo the child needs a high intervention environment. Use of the Polya model as an instructional framework for Logo activities provides just such an environment at the same time the child remains at the center of the learning activity. This paper describes two high intervention projects based on the Polya model: introduction to the Logo primitives using mazes, and writing a class newspaper.

Mazes are especially well suited to problem solving instruction using the Polya model: both goals and progress towards them are well defined. Students can easily monitor their own efforts. Each maze program seeks to develop an understanding of a particular geometric concept as well awareness of directionality and the range of turtle movements. Figures 2, 3, and 4 illustrate sample mazes.

The POND MAZE-1 in figure 2 challenges a student to bring the turtle back to its home in the pond. Logo commands sufficient to accomplish this task are given on the screen. Manipulatives such as turtle compasses and rulers are also used. New commands may be introduced through the maze programs. Obstacles in the turtle's path demand that the turtle "jump" over them using PENUP and PENDOWN. Many variations are possible. To make a maze more challenging a student may attempt it with the turtle hidden. Other versions of POND-MAZE present students with the same basic path but only RT or LT turns, or with different starting points, or perhaps with with the whole diagram rotated 45° or 30° to emphasize the relative nature of anglular measurement. Other pond mazes contain repeated patterns as solutions, ex. REPEAT 3 (FD 10, RT 90, FD 20, LT 90).

All maze programs are open for student inspection. More advanced students may discover the pattern and write a simple program to send the turtle through the maze with one command. The instructional design of commercial packages like *The Pond,* or *Algernon* by Sunburst Software explores similar topics but the Logo mazes described are much more flexible and dynamic.

Figure 2

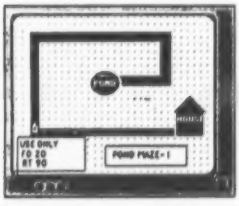


Figure 3

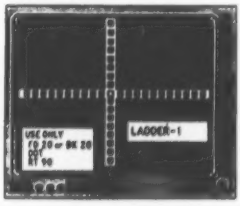
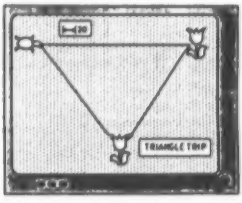


Figure 4



The LADDER maze in figure 3 invites students to make their turtle leave a mark in each of the boxes shown. The commands FD 20, RT 90, and DOT are given. After the first use of the DOT command the turtle disappears from view. Students must find a way to keep track of the direction of the turtle without the aid of a visual clue. They must use the Polya model to develop a strategy to solve this informational problem. In case of difficulty students can enter SHOWTURTLE.

The TRIANGLE TRIP maze in figure 4 sets students the problem of feeding their turtles with flowers that are arranged at the tips of an equilateral triangle. They must find two commands that, when repeated three times, will bring the turtle to all of the flowers at the tips. Through this and other similar activities beginning students experience the equal nature of the angles and sides in the equilateral triangle. Other tesselating triangle mazes introduce students to the relationships of repeating patterns in the plane. A family of mazes named OZ and

STARWARS introduce beginning students to more complicated geometric figures and more advanced angular and linear measurements. The REPEAT command, and writing procedures are introduced as compact ways to solve pattern mazes.

Vocalization and paired problem solving is encouraged. Students role play the "Knower" and the "Questioner". When the pair needs to clarify decisions, the questioner asks the knower "What do you really know so far?". The questioner attempts to guide the knower through the steps of the Polya model searching for something missing or amiss. The role of questioner is not that of a teacher, the goal of the questioner is to clarify and keep the pair on task, not to teach. During any problem solving session, students often reverse roles. The use of the knower/questioner strategy is the first step towards careful self-monitoring that is the mark of a good critical thinker and problem solver.

Having mastered the Logo primitives and the problem solving style fostered by the Polya model students begin their own projects. Many choose representational themes; not surprisingly, some of them decide to build intricate mazes. All student projects must be documented by Logo Project Planners that set out problem definition, planning and strategy choice, description of step by step execution, and problem verification and possible alternate solutions. After several successful projects students begin to see a pattern developing. The thread that binds each project solution is the method used to solve each problem. Much the same way that the Logo turtle permits students to experience the abstract world of geometry in a semi-concrete way, the patterns formed by solving problems with the Polya model aid students to heighten their awareness of organized thinking in a simple and ordered fashion.

Writing also demands organized thought. This is especially true for journalistic writing. Interviewing, editing and publishing follow clearly defined guidelines that work well with the Polya model. Students define the writing problem with the "Who, What, When, Where, and Why" method. The pyramid style of journalistic writing forces students to plan their stories from the specific and essential to the general and supportive. Clarity is a prime concern. Editing may also be seen as a problem solving process with standards of clarity and limits of space providing a framework.

Throughout this publishing process each student can fulfill the roles of knower and questioner on different levels, as reporter, editor, and publisher. At each stage students use the Polya model to write and evaluate their work. With so many built in checks and balances, the newspaper project serves as a stimulating example of self-monitored problem solving in which students can not only see but govern their own success.

The student, not any program or computer, must be the active focus of any effort to teach critical thinking or problem solving. Logo and a basic word processor such as *Bank Street Writer* are all that is needed. The Polya model is simple but not simplistic. It is widely applicable to other subject areas. Projects in Logo graphics and a school newspaper provide activities in problem solving ranging from simple, concrete editing to sophisticated exploration and manipulation of abstract ideas. Our approach, setting Logo instruction within a

critical thinking and Polya problem solving framework, encourages students' cognitive development as well as their personal involvement with computers and active problem solving.

For more information about Logo and problem solving with the Polya model, please contact:

George Collison
Computer Coordinator
Holyoke Public Schools
500 Beech Street
Holyoke, MA 01040

Logo Learning Tools for the Classroom

Dorothy M. Fitch
Terrapin, Inc.
222 Third Street
Cambridge, MA 02142
© 1987 Dorothy M. Fitch

Logo	=	A sophisticated language in which students of all ages can explore concepts within a variety of environments and structures.
Learning	=	An activity in which students gain knowledge by collecting, organizing, analyzing, and comparing facts and information in order to explore relationships, make generalizations and find exceptions.
Tools	=	Programs designed to accept data and manipulate it according to a set of rules determined by the user and the programmer.
Classroom	=	A place in which students interact with the above activities and concepts.

These terms combine some very powerful concepts relating to the use of computers in education. The challenge is to link them together in a way that will allow the best of all the parts to emerge.

I have long been intrigued by Logo—the open-ended learning opportunities it can provide, its structure as a programming language with an understated capability for serious programming endeavors, and mostly by the sense of excitement and satisfaction I have felt in many long sessions spent exploring its nooks and crannies.

As an educator, I understand that learning must be more than just the memorization of facts. A teacher must make information meaningful through organization, association, questioning and illustration. Jigsaw puzzle pieces of information must be interconnected across many areas of knowledge to create a larger recognizable picture.

But how does this relate to Logo? I imagined a way to help students deal creatively with factual information, become familiar with real-life computer applications, and learn to use the computer as a tool, while remaining in a Logo environment. The students would be able to enter their own information in any area of learning or interest and generate results to help them look at their information in a new way. These results would help them present their research in an organized manner, allowing them to uncover interesting relationships and ask new questions about their findings. An "open" language, Logo would allow students and teachers to look at the programming instructions and procedures and learn new ways of using Logo and dealing with words and lists.

The Logo Data Toolkit package is designed to achieve these goals. The four tool programs can be used by teachers or students to store information and present reports. Information generated by one tool can often be used by another. Following is a description of the tools and how they might be used to add a new dimension to learning in the classroom. The presentation will demonstrate a method of incorporating these tools in an integrated classroom project.

SURVEY

You may have had an opportunity to take the "Logo Users' Survey" prior to this presentation. This computer questionnaire, previously entered using the MAKE A SURVEY program, consists of several questions about the background and interests of conference attendees. The TAKE A SURVEY program allows an unlimited number of people to answer the questions, then records the answers. A printed report showing the percentage of people selecting each answer can be generated at any point.

The Survey program provides students with a tool for studying popular preferences, trends and attitudes through the use of survey questions. Whether the subject concerns environmental issues, political views, personal habits or favorite ice cream flavors, students will broaden their outlook on the world, develop their own value systems, and gain an awareness of their own individuality.

Teachers can use the Survey tool with students to learn more about their students' lives at home and present new information in an informal way. Students can learn to conduct research and find out more about their classmates. Results generated by the Survey tool can be represented in chart or graph form using a separate conversion program.

DATABASE

Using the Database tool, a student or teacher can design a database format or structure, enter data into categories, modify or delete records or categories, search and sort a database, and design a report format for printing data.

A database can store information about everything from rock bands to presidents. As students access data, they may find relationships that might have otherwise escaped their (and their teacher's) notice.

A three step method of introducing students to the concept of a database program will help them learn to use this or any other database tool.

1. Allow students to search and sort an existing database. Develop questions for them to answer as they use the database. A database of explorers might prompt questions such as "Which country produced the greatest number of explorers?" or "Which explorer traveled the greatest number of miles?"

2. Next, have students enter data into a previously designed database structure. Student groups can research and enter data about each planet. To help fellow classmates select a good book, they can enter information about the books they have read.

3. Finally, teach students how to design a database structure by entering category names and designating each category as either numeric or nonnumeric. Now the students can complete a database of their choosing from start to finish, computerizing their hobbies and interests.

Researching, organizing, storing and looking up information using a database can be a powerful activity in virtually any subject area. The Logo Data Toolkit takes data management one step further. Information stored by the Database tool can be converted to be used by the Charts or Associations tools.

ASSOCIATIONS

The Associations tool allows the teacher or student to set up pairs of related words or phrases to be matched in a game or quiz format. Designed as a drill on facts or a trivia game, the associated pairs can involve countries and their capitals, foreign language words and their English translations, inventors and inventions, books and authors, mathematical expressions and their values, or students in the class and their birthdays.

The Logo Data Toolkit allows these word pairs to be matched in either a game format, with multiple attempts permitted, or a quiz in which student percentage scores are stored. These scores can then be managed through a set of hidden teacher commands. Association games or quizzes can be created by converting data stored in database format.

CHARTS

Learning to prepare and read charts and graphs is a typical math class activity. This Logo Data Toolkit program will draw a variety of charts and graphs based on information that is either entered directly or converted from the Survey or Database tool.

Pie charts, colored pie charts, line graphs and bar charts can all be generated using the same data. A printed report displays the numeric value for each entry as well as the percentage of the total number involved.

Students researching various areas can use charts and graphs to illustrate their findings. For example:

Science	A line graph showing the growth of a plant or organism
Social Studies	A bar chart comparing county populations or areas
Business	A pie chart displaying amounts of television advertising
Math	A line graph charting a number series
Economics	A line graph illustrating stock market fluctuations

SUMMARY

The Logo Data Toolkit offers a creative use of the computer in the classroom. Students and teachers can review and adapt the tools to meet their own needs. For both the beginning and experienced user, the programs provide an opportunity to learn to use the power of the computer as a tool. Teachers and students alike can approach old subjects with new enthusiasm.

Creating a LOGO Environment in a Geometry Class

America H. Foster
George Washington Carver High School
Fieldale, Virginia

For the decade of the 80's, the National Council of Teachers of Mathematics (NCTM) is emphasizing problem solving and computer applications in mathematics. In geometry, LOGO is an excellent tool for problem solving, because the microworld of the turtle is a perfect geometric field. The LOGO language seems to motivate students to explore and to experiment, which usually produce lots of creativity and critical thinking.

Therefore, this project, "Creating a LOGO Environment in a Geometry Class", is based on a problem solving approach. It consists of lesson plans, reference sheets and original LCSI LOGO programs written to enhance the study of high school geometry. The NCTM Student Merit Awards Program is also included; "it provides enrichment material in a convenient format and a way to recognize student achievement in mathematics". (Sachs, 1984, p. 3)

The primary objectives of "Creating a LOGO Environment in a Geometry Class" are to teach the LOGO language and to guide the students toward more analytical thinking through problem solving activities. The main activities are laboratory experiences and projects: the laboratory experiences are for the students to explore and generate new ideas; the projects allow for organizing and developing modularity in structured programming.

I have chosen the following topics with selected activities to explore: the history of mathematics and computer applications, basic properties and concepts of geometry, diagrams of formal proofs, classification of triangles, similarity, the Pythagorean Theorem, regular polygons and patterns, robotics and computer art, symmetry in transformations, area and volume measurement, geometric constructions, coordinate geometry, and the conic sections.

As an introduction of computer applications to the History of Mathematics, students learn to use a word processor to write reports on various historical and computer topics from the NCTM merit programs.

As basic properties are studied, students use LOGO 'make' statements from list processing to teach the turtle how to generate if-then conditional statements through random selections of parts of speech. Students then identify the hypothesis and conclusion, write the converse, inverse, and contrapositive, and evaluate the truth of each statement.

In writing formal proofs, a diagram which represents the given, is needed. Students teach the LOGO turtle to make these diagrams, which usually consist of points, lines, planes and polygons.

As students classify triangles according to their sides and angles, the turtle is taught how to draw acute, obtuse and right triangles; scalene, isosceles and equilateral triangles. To add a little spice, measurement is introduced by making the turtle draw special triangles with given lengths and angle values. Structured programming is introduced here by combining all of these triangular procedures into a larger calling procedure.

The concept of similarity is seen graphically when the students use variable procedures to make a given figure change size, but not shape. This helps to distinguish between congruent triangles and similar triangles.

Students learn to prove the Pythagorean Theorem, the most popular geometric theorem of all, through nonconventional ways: instead of using squares on the legs and hypotenuse, semicircles and regular polygons, such as triangles and hexagons are used. The art of making the turtle draw these diagrams and use output statements for the area measurements will challenge any student and teacher alike.

To study regular polygons, students play turtle on graph paper first. Then they draw the polygons, identify patterns, and write procedures using the 'repeat' command and variables. Mathematical concepts of interior and exterior angles are developed as generalizations are made.

The aesthetic side of geometry is always enjoyed. By using robots, computer art designs with polygons can be very colorful and creative

as well as mathematical conceptual. The Tasman Turtle Tot is excellent for problems involving angles. Even though he is bounded by a 21-foot umbilical cord, he makes the screen turtle come alive when studying spacial concepts. We found the use of Big Trak very effective in clarifying directions of right and left, especially when he or the turtle is facing the student.

Transformations are very interesting to many students as an introduction to mapping; thus, bridging the gap between algebra and geometry. We usually start with symmetry because of its aesthetic side. Symmetry in nature also brings out the beauty of geometrical shapes and sizes. Students study line and point symmetry of things found in nature and teach the turtle to make them on the screen.

Measurement of solid geometric figures is also explored through output statements. Some students find it hard to sKetch the 3-D figures that are needed to visualize the dimensions when finding surface area and volume. Therefore, some of these figures are drawn using variable procedures and stored on a diskette for others to use. In plane geometry, medians, altitudes, and angle bisectors are drawn on special figures, such as equilateral and isosceles triangles; the areas are then calculated through the use of formulas and output statements.

Geometric constructions always seem to fascinate students. Their favorite ones are the SSS, SAS, and ASA constructions of congruent triangles. These constructions usually give them more respect for angles. The LOGO turtle is taught how to make these constructions in the microworld of the computer.

In coordinate geometry, the turtle command of 'setpos' is used to make the turtle plot coordinates on the screen. Students are usually given sets of ordered pairs to plot pictures. Then they create pictures on graph paper, write procedures for each separate part, and put them all together under a calling procedure. In a more advanced problem solving application, a computer version of the scrabble word game was developed in BASIC by two students for a gifted-and-talented project; we then proceded to write LOGO procedures for the game to fit into the turtle's world.

Some other coordinate geometry projects that we are now exploring are slopes of lines and string art. We also hope to teach the turtle how to graph the conic sections. "Creating a LOGO environment in a Geometry Class" is definitely incomplete, for as new concepts are being introduced and studied in geometry, we try to find ways to enhance them with LOGO.

The advantage of this project is that students seem to develop a better understanding of the abstractness of mathematics. The imaginary becomes reality to them through the movement of the turtle. This process of growth and development, however, does not come over night; but, through hours and hours of debugging and dissecting. The students don't seem to mind the time-on-task, for they are highly motivated by LOGO. As a result, problem solving abilities and strategies become more acute and a feeling of special independence seem to develop.

Bibliography

Abelson, H. (1982). Apple Logo. Petersborough, New Hampshire: Byte/McGraw-Hill.

Bearden, D. (1983). The Turtle's Sourcebook. Reston, Virginia: Reston Publishing Company.

Fey, J. (Ed.). (1984). Computers and Mathematics. Reston, Virginia: NCTM.

Lough, T. (Ed.). (1983). The National Logo Exchange. Westport, Connecticut: Meckler Publishing.

Muller, J. (1983). The Turtle News. Richardson, Texas: Young People's Logo Association

Papert, S. (1980). Mindstorms: Children, Computers, and Powerful Ideas. New York: Basic Books, Inc.

Sachs, L. (Ed.). (1984). Student Merit Awards High School. Reston, Virginia: NCTM.

Sachs, L. (Ed.). (1984). Student Merit Awards Middle School. Reston, Virginia: NCTM.

Watt, D. (1984). Learning with Apple Logo. Petersborough, New Hampshire: McGraw/Hill, Inc.

The Platonic Solids

Jacqueline M. Foster
Laurel Park High School
Henry County Public Schools

Plato, a disciple of Socrates, was one of the greatest thinkers of all times; an accomplished mathematician who was also interested in astronomy. He believed that the heavenly bodies, and their movements, were perfect examples of geometric forms. (World Book, 1975). He claimed that the atoms of the elements were in the same shape as the five regular (having all congruent sides and angles) polyhedra. While Plato was not exactly correct about these atoms, it is true that many crystals are in the shape of these polyhedra.

Plato discovered that there are only five polyhedra which have regular polygons as faces and all congruent edges. Today, these five regular polyhedra are often called the Platonic Solids. The Platonic Solids are named according to the number of faces: the tetrahedron (4 triangles), hexahedron (6 squares), octahedron (8 triangles), dodecahedron (12 pentagons), and the icosahedron (20 triangles).

Archimedes studied under Plato and wrote a book about the entire set of Archimedian Solids. Five of the thirteen solids are truncated counterparts of the Platonic Solids. By 'truncated counterparts' I mean that the corners of the Platonic Solids were cut off, thus producing new figures. The use of the Platonic and Archimedian Solids to bring out mathematical concepts is the basis of my project, which deals with the integration of the LOGO computer language as a tool into a science project.

The project consists of drinking-straw and tagboard models of the regular polyhedra, display charts, solid figurines taken from the Dungeon & Dragon game, and an illustrated booklet of related mathematical concepts. Computer programs, written in LCSI Apple LOGO II, are used to generate printouts of these solids and their Archimedian counterparts.

Some of the mathematical concepts that were studied are symmetry, congruence, variables, constants, coplanar, non-coplanar, interior and exterior angles, pattern formations, Euler's formula, and topology.

The concept of symmetry can be illustrated using any of the Platonic Solids, because all of the regular polygons that form each solid have lines of symmetry. The triangle has three, the square has four, the pentagon has five, and the hexagon has six.

Congruence can be illustrated with any of the the five solids, because they are all regular. The net of the dodecahedron is used to illustrate this concept.

Variables and patterns are used in writing programming procedures that can be used with figures of different sizes. Once the pattern is discovered, one variable procedure can be written that will draw an infinite number of designs just by changing a value.

The coplanar concept can be illustrated with the hexahedron and the tetrahedron. The hexahedron has three faces and the tetrahedron has four. If each were used as a stool, the tetrahedron would give the best support because it rests only in one plane; whereas, the hexahedron with four sides, rocks back and forth on four different planes.

The relationship between interior and exterior angle measures is brought out through the construction of the patterns. The computer was used primarily as a tool to generate nets for these constructions to enhance this project. However, the computer printouts of the nets were not balanced correctly because of the pixels in the CRT; so the patterns had to be made by hand using a ruler, compass, and protractor.

Euler, a famous Swiss mathematician, developed a formula which shows a relationship between the number of faces (f), vertices (v), and edges (e) of regular polyhedra. The formula is as follows: F + V = E + 2. This formula illustrated the concept of pattern formations.

The map coloring problem, a concept of Topology, is illustrated through a multi-colored icosahedron. The objective is to color this polyhedra using the least number of colors so that no two adjacent faces will have the same color.

"The Platonic Solids" was a science project done for our local Science Fair. It won first

place in the category of Mathematics at my local high school and in our school division. It was also one of six projects selected to be entered in the Western Virginia Regional District Science Fair at Roanoke College last April. There, "The Platonic Solid" won awards from NASA, the U. S. Army, and The National Council of Teachers of Mathematics. The project placed second overall in the category of Mathematics and Computers.

And finally, upon entering the computer part of "The Platonic Solids" in the Judges' Choice category of the '86 Merit Competition Awards Program sponsored by the Apple Computer Clubs International (ACCI), it was awarded a Second Place honor on the national level. The key factor to its merit was that it used the computer as an integrating educational tool into other subject areas, and not just in and of itself.

Bibliography

Abelson, H. (1982). Apple Logo. Petersborough, New Hampshire: Byte/McGraw-Hill.

Papert, S. (1980). Mindstorms: Children, Computers, and Powerful Ideas. New York: Basic Books, Inc.

Sachs, L. (Ed.). (1984). Student Merit Awards High School. Reston, Virginia: NCTM.

Wenninger, M. J. (1985). Reader Reflections. Mathematics Teacher, 78, 12.

Wenninger, M. J. (1985). Reader Reflections. Mathematics Teacher, 78, 90.

Wenninger, M. J. (1985). Reader Reflections. Mathematics Teacher, 78, 166.

Wenninger, M. J. (1985). Reader Reflections. Mathematics Teacher, 78, 238.

Wenninger, M. J. (1985). Reader Reflections. Mathematics Teacher, 78, 330.

World Book Encyclopedia. (1975), Vol. A.

World Book Encyclopedia, (1975), Vol. P.

Reading, Writing, and Logo: The Ultimate Language Experience

Pamela Freyd, Ph.D.
Graduate School of Education
University of Pennsylvania

Because turtle graphics give new users easy access to the computer language Logo, many educators have overlooked the fact that Logo is also a powerful and expressive language that allows for the serious manipulation of natural language. Just as computers whose origins are based in number crunching are now used by most people for processing words, so Logo designed as an environment for exploring mathematical concepts may find greatly expanded acceptance as a tool with which children can explore natural language. By its very nature, Logo provides an environment in which children are encouraged to discuss and solve problems, write and revise their solutions, read and share their results. Logo is a language oriented environment. My talk today has three parts: a brief background of the Logo teaching experiences that have convinced me of its importance for language arts; a description of some ongoing research using Logo to help teach revision processes; and detailed descriptions of four Logo activities designed to encourage natural language exploration.

For three years I have been investigating Logo with children in a computer lab in a K - 8 public school in downtown Philadelphia. The excitingly diversified 920 children came to the lab with their classes while their teachers were provided what is called in Philadelphia "prep-time", class preparation time. Clearly not the ideal computer or Logo teaching situation, we made it work through the extensive use of volunteers and student peer tutors. Parent support was critical for our success: the parents raised the money for 34 Commodores and helped to run after-school programs to attract volunteers for the children's classes.

Before becoming a full-time computer lab teacher, I had been an 8th grade English teacher and an elementary classroom teacher. My dissertation research was in the field of child language acquisition. A year in a regular classroom with BASIC and one disk-driveless Apple II, however, convinced me that computers might have a profound effect on classroom organization and teaching methods. That lead to Logo and to the start of the computer lab. Like many teachers, I thought of Logo in terms of its mathematical potential. It was the students themselves, especially the hearing-impaired, who helped me see the language potential.

The hearing-impaired came as a class to the lab. Although I had some sign and fingerspelling and they had some lip-reading skill, guiding the class was a challenge. Modeling exercises and activity cards got us started but could not support the individual directions and questions that soon followed. The obvious answer was staring at us from the split screen. With no interference to the graphic work at hand, we soon began "talking" with each other all the time using the lower text lines. Nothing remarkable in itself, but something remarkable happened to the quality of our communicative relationships. Logo provided both a stimulus and a tool for communicating about all kinds of things that would not have been possible with this particular group without it. The computer with Logo was a medium of communication between people.

The natural next step was to begin to use Logo in this way with all the children. Soon older children were making illustrated computer stories for younger children combining simple graphics and print commands. Although our main projects were aimed toward developing mathematical skills and graphic control, in fact we moved more and more in the direction of Logo words and lists.

Logo fits naturally into an integrated reading-writing curriculum in which the educational philosophy is directed toward reading and writing experience as communicative experience rather than as subsets of discrete skills. At a beginning reading level this curriculum would include a language experience approach (LEA). Any version of Logo can be easily and successfully used to write stories with children. The advantage of using Logo over a word processing program at this level is that pictures can be used to encourage writing and then easily combined with text. Logo is an easy tool for teachers to use to plan activities to reinforce either sight vocabulary or spelling words for young children.

At an intermediate level, language skills such as recognition of parts of speech can be meaningfully embedded in programs that play with words (sentence generator programs). Children can get direct and immediate experience into the nature of audience by writing interactive programs. The fact that language between people is cooperative can be made explicit for children by contrast with computer language which does not support ambiguity or slight variation. Pointing out parallels between writing programs and composing stories supports improved composition structure (Newkirk, 1985). Writing and programming involve the same major stages: prewriting and planning, composing, revising, editing, and publishing. Many of the same considerations must be taken into account. For example, both the writer and programmer must have in mind who the reader will be and what is his/her background. Both the writer and programmer must decide on purpose and focus. The most important similarity between programming and composing is that both seek to communicate ideas clearly and concisely.

The area of revision is one in which Logo has the potential to offer great support to language programs. The current momentum in writing instruction is toward a process rather than a product approach, and Logo is very much a process environment. Debugging is not felt to be an unpleasant task in Logo. In fact it is often said that 'bugs should be cherished' because with bugs we can learn from our mistakes. Debugging in Logo is parallel to revising a composition. Revision as part of writing processes has been little studied, however, because before computers and word processing, the physical effort involved in revision was disheartening. Research which is just now becoming available (Daiute, 1985; Kahn, 1986) points to the fact that children's approach to revision is local rather than global when, in fact, revision is even attempted. That is, children most often fail to see how making changes in one part of a composition will affect other parts of the writing.

We have been conducting research with 4th and 6th graders to see if direct instruction in debugging purposes, strategies and techniques will prove to have any influence on revision attitudes and practices. Indeed, we are aware that early expectations that children would naturally do lots of debugging in Logo have been every bit as unrealistic as teachers' expectations that children would naturally revise if given word processing. Logo, however, presents a special challenge or (we believe) opportunity in this area. In Logo children are working in two modalities, visual and verbal. Because it should be relatively easy to see how a change in any one part of a picture affects the whole picture, we believe that children can be led to a greater understanding of the purpose of revision. By helping children see how Logo debugging affects a total picture in the graphic domain, we hope that children can be helped to see how revisions in writing inevitably affect a whole composition in a verbal domain. This is just one of many new ways that Logo offers us to look at traditional language experiences.

The types and numbers of Logo activities that can help children learn about natural language are as open ended as Logo itself. I am going to present four activities that I have found to be very successful when they are suitably integrated into langage curriculum.

1. Create Mad Libs that fit the word classes students are studying. (Exploring parts of speech.)

2. Create word lists and using them to make random sentences. (Exploring word order.)

3. Write branching stories in Logo. (Creative writing.)

4. Create sentences to test subject-verb agreement. (Exploring grammar.)

References:

Birch, A. (1986) The Logo Project Book. Cambridge,MA:Terrapin.

Daiute, C. (1985) Writing & Computers. Reading,MA:Addison-Wesley.

Dickson, W.P.(1985 May) Thought-provoking software: juxtaposing symbol systems. Educational Researcher 30-38.

Kahn,J. (1986) Paper presented at Keystone Reading Association Convention, Philadelphia, PA. Oct.

Lesser, M. (1985) Logo for Micros. Kent,England:Butterworth.

Newkirk, T (1984-85 Winter) Writing and programming: two modes of composing. CRLA. 40-43.

Perkins, D.N (1985 Aug/Sept) The fingertip effect: how information-processing technology shapes thinking. Educational Researcher. 11-17.

Logo Music Tools
"MusicLand"

Applications in Music Using Logo

James S. Fry
Novi Community Schools, Novi, Michigan

MusicLand was developed by Sylvia Chang, Kingswood School, Bloomfield Hills, Michigan, James Fry, Novi Community Schools, Novi, Michigan, Austia Hollen, Southfield Public Schools, Southfield, Michigan, and Lary Smith, Wayne County Intermediate School District, Wayne County, Michigan. MusicLand was developed as part of a two county Logo integration project. The goals of the project were to develop Logo activities that could be used by classroom teachers, with little or no Logo experience, in the different curriculum areas. One of the groups that formed was a music group. This is the group that developed the MusicLand Microworld.

MusicLand is a series microworlds created to help explore the elements of music and sound through the Logo language. The MusicLand microworlds provide structured music activities or games along with a set of "tool" procedures to explore and discover many areas of music.

There are two major parts to MusicLand. The first is a more structured set of activities called **"Music.Games"**. Music.Games consists of 3 different "games"; **High/Low, Long/Short,** and **Up/Down**. These games are aimed at the Kindergarten to second grade levels. The objectives for the games are to distinguish between high and low pitches, long and short tones, and upward and downward movement of notes in a musical line.

The second part of MusicLand consists of a set of "tools" grouped into **"Write.Tools"** and **"Sing.Tools"**. These tools have a set of activities that are much more open-ended. The tools allow you to work with **"tuneblocks"**. Tuneblocks are short phrases or lists of musical notes and durations. MusicLand tools allows one to write a "tuneblock", sing it (hear it played by the computer), and manipulate it many different ways or combine many tuneblocks into a song. The many operations that can be performed on a tuneblock are; **Lengthen** or **Shorten** (the use of augmentation and diminuation), **Reverse** and **Mirror** (played backwards or inverted vertically around a tonic or home tone), and **Transpose** (to raise or lower by steps or octaves). These "tools" can also be combined to create many new sounds from one original tuneblock.

Similar to Seymour Papert's idea of a "Mathland" as described in his book Mindstorms, the microworld, "MusicLand", could be set up in the music classroom. Separated from the rest of the classroom by book shelves and a computer cart some students working in groups of two or three could explore the elements of rhythm, melody, and form at the computer while the other members of the class could be involved in their lessons.

All the MusicLand activities are structured in the same way to give a beginning user of Logo or the MusicLand microworlds a guide to follow. Each lesson consists of the following major topics; Topic presentation, Pre-computer Lesson, Guided Discovery/Logo Activity, Follow-up Off-computer Activity, Extended Learning Activity, and Notes to the Teacher.

MusicLand Microworlds

High/Low Game

Objectives:
The student will distinquish between high and low pitches.

Prerequisites:
As preparation for this game, students have experienced high and low sounds through speech (i.e., animal sounds, singing, and movement).

Background information:
Grade level K-2

Logo activities:
Students will work in pairs at the computer. With the High/Low game loaded, the students will take turns typing and listening. One student will type the letter "p" followed by two inputs between 0 and 11. Both inputs are note numbers to be played. The student listening will tell the typist which note is high and which is low. Other games or variations can be played or tried by the students.

Long/Short Game

Objectives:
The student will distinguish between long and short tones.

Prerequisites:
As preparation for this game, students have experimented with long and short tones through instrument exploration, movement, speech, and singing.

Background information:
Grade level K-2

Logo activities:
Students will work in pairs at the computer. The same type of activities will go on as in the High/Low game. One student will type while the other listens and tries to tell which tone is longer and which is shorter. The typist will type an "L" and two inputs between 1 an 10, representing the durations of the tones. Other games or variations can be played or tried by the students.

Up/Down Game

Objectives:
The student will distinguish between upward and downward movement of notes in a musical line.

Prerequisites:
The students will have sung, played, listened, and moved to ascending and descending melodic lines. This activity provides yet another opportunity to hear randomly generated sequences of ascending and descending pitches.

Background information:
Grade level K-2

Logo activities:

Students will work in pairs at the computer. Taking turns typing and listening the students will type "N U" for ascending pitches and "N D" for descending pitches. The student listening will hear a randomly generated set of five pitches twice. They must then tell whether the direction is up or down.

Write and Sing Tools

Objectives:

The students will play simple familiar melodies by ear on the computer by using WRITE and SING.
The students will identify repetition and contrast in musical blocks (motives).
The students will use augmentaion and diminutation to vary a given melody by using LENGTHEN and SHORTEN.
The students will use REVERSE (tuneblock played backwards) and MIRROR (tuneblock inverted vertically around tonic or home tone) to vary a melody.
The students will construct major and pentatonic scales by ear on the computer and use the TRANSPOSE tool.

Prerequisites:

The students should have had the opportunity to play familiar melodies on classroom instruments (piano, recorder Orff instruments, resonator bells). Also musical repetition and contrast through singing, playing, creating, and listening.

Logo activities:

The students will work in various size groups to experiment and explore the many different activities in the Write and Sing Tools. They will compose and save tunesblocks. They will then have the opportunity to manipulate those tuneblocks many different ways with the collection of musical tools. The students will be able to explore how the tools can be combined to create new effects with their tuneblocks.

Procedures

Below is a listing of some of the "tools" that are used to manipulate the tuneblock lists in the MusicLand microworld. They all perform the same basic type of operations. Since tuneblocks are made of a List of two lists, (note list and duration list), each of the tools is designed to perform some set of operations on one or both of the lists and output that information, or new list, to the main procedure, SING, that then plays the notes at the desired duration.

```
TO TRANSPOSE :PHRASE :SHIFT
    MAKE "FPHRASE FIRST :PHRASE
    OUTPUT LIST (TRANSPOSE.NOTES :FPHRASE :SHIFT) (LAST :PHRASE)
END

TO TRANSPOSE.NOTES :FPHRASE :SHIFT
    IF :FPHRASE = [] OUTPUT []
    OUTPUT SENTENCE ((FIRST :FPHRASE) + :SHIFT) (TRANSPOSE.NOTES
    BUTFIRST :FPHRASE  :SHIFT)
END
```

```
TO REVERSE :PHRASE
    OUTPUT LIST (REVERSE.LIST FIRST :PHRASE) (REVERSE.LIST LAST
    :PHRASE)
END

TO REVERSE.LIST :LIST
    IF :LIST = [] OUTPUT []
    OUTPUT SENTENCE (LAST :LIST) (REVERSE.LIST BUTLAST :LIST)
END

TO MIRROR :PHRASE :TONIC
    OUTPUT LIST (MIRROR.LIST FIRST :PHRASE :TONIC) (LAST :PHRASE)
END

TO MIRROR.LIST :LIST :TONIC
    IF :LIST = [] OUTPUT []
    OUTPUT SENTENCE (2 * :TONIC - FIRST :LIST) (MIRROR.LIST BUTFIRST
:LIST :TONIC)
END
```

East Coast Logo Conference '87

METHODS OF SIMULATING A ROBOT'S BEHAVIOR USING LOGO

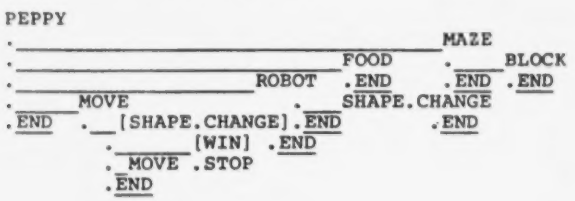
MICHAEL A. GELMAN
Packanack School, Wayne, NJ

I. THE ROBOT

I took my idea from a do-it-yourself robot named Peppy, which travels more or less straight forward until it hits something or until a distinct sound is made. The robot then executes a 90° to 270° turn and continues forward. I was experimenting with the robot in a labyrinth and it was taking too much time. I decided that my testing would be faster if I made a computer simulation. I decided on Logo as my language, namely LogoWriter, and PEPPY was born.

II. THE PROCEDURES

The Procedure Tree



III. THE PRINCIPLE

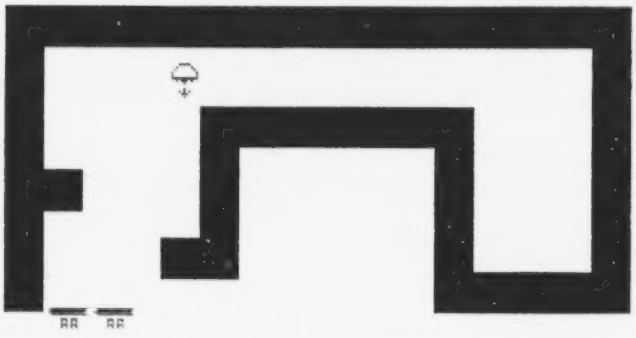
The basic structure of the program is the PEPPY top program. MAZE tells where to put the BLOCKs. FOOD draws the two AA batteries at the exit. ROBOT places Peppy in the maze. MOVE is the recursive program that determines if Peppy wins, keeps on going, or bumps into a wall. In the latter case, MOVE determines how far Peppy will turn RANDOMly. MOVE also keeps track of the number of times it is run, because MOVE is a recursive program. WIN displays "Congratulations" on the screen and the number of times MOVE was run in the command center. However, if MOVE is run more than 1523 times, the program prints "Slowpoke!" and stops

Copyright Michael A. Gelman 1987

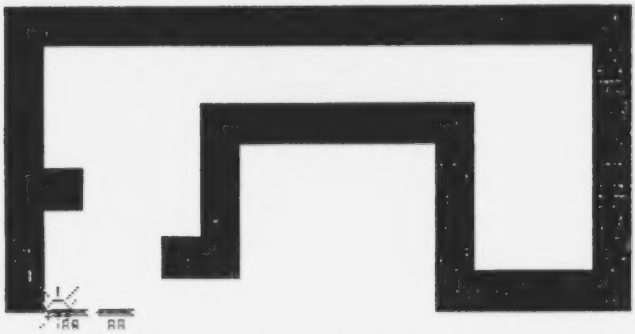
automatically. Why 1523? It was the average number of moves for Peppy to get through one of my mazes.

IV. THE CONTEST

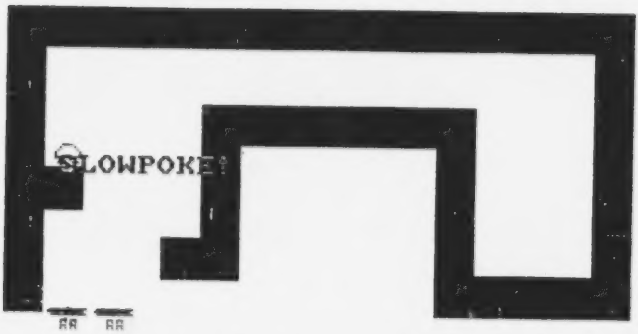
I propose a "Make Your Own Maze" contest to create a Peppy maze with the lowest average number of times for the robot to get through it. The current top record average is under 1500. The "Slowpoke" rule will be in effect during the contest, with the threshold being progressively lower. Those with more than half of their trials having "Slowpoke" results will be eliminated until all are "Slowpokes" but one. That one will be declared the winner.



Peppy working its way through a maze



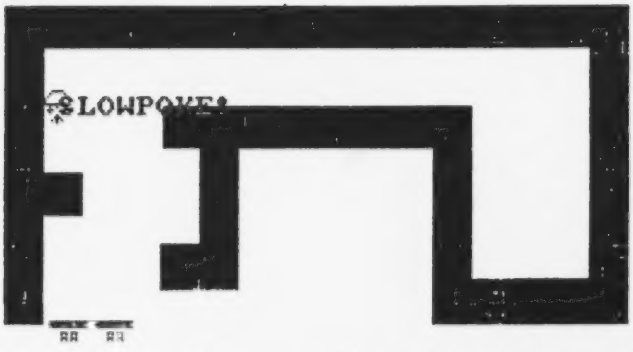
Peppy getting its new batteries



Peppy being a slowpoke

The Lighter Side of Peppy

"Ouch!"



"Whadaya mean? I only get to go FD 4!"

Congratulations!

"Sorry, Duracell, I'd rather have an Energizer."

Logo Quickly

by Jonell Glass
and Andrew Evans

Teaching Logo in a camp setting poses challenges different from those presented in a classroom. The biggest constraint on a camp counselor is time: we have two weeks to give our campers quality experiences with Logo. To provide these experiences, counselors circumvent traditional Logo teaching methods and jump right into programming. Without a strong background in Logo, however, children often seriously misunderstand its basic programming concepts. Preventing these misconceptions but also giving children a valuable experience with Logo can be a real challenge. We have developed various strategies to help both beginning and advanced campers use Logo efficiently and effectively.

JONELL: Beginning campers - many of whom have used computers for only recreation before - find the whole concept of programming foreign. Introducing these children to computing logic requires concrete examples of each primitive; however, ten three-hour sessions mandate that the examples be precise. Each child's introduction to the computer lasts only one day. In that day, the child uses 20 or more primitives and may even write procedures. Because of the intensity of this introduction, we expect children to forget some points and be confused by others. To help them out, we place flash cards, poster boards, and other visual aids listing primitives and their meanings throughout the classrooms. However, each child usually focuses on one aspect of the introduction. Fascinated by what he sees, he repeats one series of commands over and over, changing it each time. Counselors encourage this; we see it as the basis for expansion of ideas later on.

The second day of camp, the counselors pick up on each child's fascination and use it to clarify the points the child has forgotten or confused. For example, the camper who wants to draw spaceships will learn to utilize the formulas for blocks and triangles in each aspect of his program. The child who likes designs will combine repeat statements and basic shape formulas to create those designs. Originally, the children write each command down in a notebook as they type it into the computer; these are then translated into procedures and saved on a disk to be refined later in the session.

When children tire of learning new concepts, we reinforce angles and distances by playing games. Angle Invaders and mazes, reinforcing magic numbers and spatial relations respectively, are two of the more popular versions of these. Challenging children to get the answer on their first try but giving suggestions if they have trouble keeps the children playing, and learning. Children also frequently revert back to making simple designs; again, we suggest that the camper use additional programming techniques in those designs, so they practice using as many primitives as possible during their "free time".

Because the campers have so little programming time, they must learn that procedures can be used in more than one program. Each counselor has devised his own way of teaching this; mine is constructing a file of polygons, two procedures for each shape, one using right turns and the other left. This "shape file" is revised as the children grasp repeat statements and variables. Once the campers learn to use combinations of variables, the file shrinks to just two procedures, using right and left turns, which input the number and length of sides to draw any polygon. This file becomes the crux of the child's programming, because it serves as a foundation for all his pictures and graphics games.

Once this file is created (not necessarily before it is revised), the campers begin to incorporate the shapes into their picture procedures. The children replace the primitives in the procedure with the proper shape procedure whenever possible; they place the remaining commands in other procedures until the picture is drawn by a series of procedures called upon by one master procedure. Although we arrive at that procedure via a roundabout

route, the campers begin to see the true importance and power of procedures. Finally the camper quits programming backwards and starts programming forwards, conceptualizing the final product first and using short procedures to create it all along. Revisions, additions to the program, and trouble shooting become much more common easier when the program is broken into procedures.

The most valuable technique in teaching beginning Logo quickly is repetition. The children write procedure after procedure, draw picture after picture, in a similar manner. Once the campers understand the format and can utilize it effectively, we show them how to generalize that format. For example, every shape in my campers' shape files uses the variable :SIZE to represent the length of the side. Because of this, the children often think that :SIZE is the only variable Logo will accept. When I feel they really understand variables, then I change the variable to their names for one procedure, my name for another procedure, and an inanimate object (such as :SWIMMING.POOL) for a third procedure. This method of introducing variables prevents much confusion; the same technique works to teach the other basic concepts in Logo, too.

ANDREW: One of the greatest challenges I was faced with as an instructor at TIC Computer Camp in Washington, DC, was that of designing coursework to teach older children (sixth grade and up) elementary topics in computer science and artificial intelligence programming. After trying unsuccessfully to use Pascal for the task, I soon discovered that Logo, which we were using with the younger campers, integrated the programming structures necessary for my undertaking (such as recursion, local variables, and list-processing) with unprecedented user-friendliness and ease of use. These factors were very important to me, considering the time constraint under which we camp counselors had to operate. I had to be able to get these kids through my course in just two weeks, and I didn't want to have to spend a lot of time getting them acquainted with a new language and operating system. In addition, the graphics capabilities would help bring to life the difficult subject matter I was attempting to present to such young students. At first the older students balked at being given Logo to work with, dismissing it as "that kindergarten language." They felt they were being demoted when they were put in my group, and many of them clamored for Pascal. But after I explained that we would be using functions of Logo that even the mighty Pascal couldn't duplicate, they agreed to give it a try.

Once I had decided on Logo as the language for my course, the coursework nearly designed itself. One of the common program examples used to teach Logo list-processing is a program that makes silly sentences by fitting randomly selected parts of speech into a simple sentence structure. This to me was an ideal starting place for deeper exploration in natural-language processing. I had my students write more complicated sentence generators in which the sentence structure was non-deterministic and based on a larger tree-structure scheme. Then I had the students organize similar sentence generators of this type into clusters. Soon, each student had built a program that could babble away merrily with a "personality" all its own. One notable program, which soon became a hit a camp, was a "Valley Girl" simulator whose accuracy was even certified by our resident experts, the female campers.

Now that the students had gained some experience with sentence structure manipulation, they were ready to start doing the whole thing in reverse; that is, they were ready to write simple parsers. But first, I made sure they understood recursion by working through examples. Then I started them right away on a recursive-descent parser. This is a concept that is hard to grasp, even for some college computer science majors, but I found my students to be amazingly receptive. Soon they had a drawing program, called Fred, that followed simple English commands, like "Draw a red box," or "Draw a green one under the blue triangle," or "Make a smaller white one." All the language comprehension was done by recursive-descent, and when verbal responses were called for from the program, they were supplied by sentence generators to give a less predictable, more human feel. The kids were very excited about the results. They spent additional time carefully

adding improvements, and competing with each other by adding new commands and special functions.

The results of this experiment were very gratifying to me. I decided to try a similar technique with much younger children. To my surprise, I had success with them that far outstripped my expectations, and in some cases even rivalled the results with the older children. I even had them writing recursive-descent parsers and a drawing program like Fred. None of this would have been possible without a programming environment like Logo. It has been my experience that with the right direction, students using Logo have much greater attention spans than those using languages like Pascal or BASIC. Also, these other language environments lack much of the sophistication and versatility afforded by Logo. I doubt that my work would have been possible without Logo.

Jonell Glass
205 B West Washington Street
Blacksburg, VA 24060

Andrew Evans
5141 Hagerstown Hall
University of Maryland
College Park, MD 20742

© 1987 E. Paul Goldenberg

Language Science, Language Play:
Using Logo to Explore What You Know

E. Paul Goldenberg, Ed.D.
Education Development Center, Inc.,
55 Chapel Street
Newton, MA 02160

It seems an odd diversion to talk about science at the very beginning of a paper on language, but having just published a book (Goldenberg and Feurzeig, 1987) devoted entirely to explorations in language, I feel a strong personal need for this diversion. In fact, with another recent book on the topic (Sharples, 1986) and four others that include significant amounts of language exploration with Logo (Birch, 1986; Friendly, 1987; Harvey 1986 and 1987), it seems a particularly good time to reflect on what is to be gained by computer activities with language.

One purpose, of course, is the enhancement of language study. Some things that are worth doing are easier to do with computers. But there is also a contribution that language study can make to childrens' scientific development—particularly as the structural elements of language are made more manipulable and "concrete" by computer. That contribution seems so important and is so neglected in conventional school curricula that I wanted to give it its own forum here.

Using the word *behaviors* somewhat loosely, science is the attempt to discover and describe the behaviors of naturally occurring things. A scientist might collect large numbers of plants and invent a system of nomenclature based on their gross structure, or might, theoretically or experimentally, try to discover the fundamental building blocks of matter. And a scientist might try to uncover the social rules followed by chimpanzees in the wild, or might try to determine the developmental steps in children's language acquisition.

Most of what we casually think of as the subject matter of science is inaccessible to children. Kids can't get particle accelerators or wild chimpanzees in school and wouldn't have the time, money, and mathematical sophistication to make much use of them anyway. Even if, as occurred in a classroom in which I was once substituting for my team leader, a child thinks he notices a pattern in the data that he has been memorizing about the solar system, he cannot check out his idea unless the book he happens to have contains the data he happens to need. So, for the most part, school science becomes the acquisition of facts that have been found by people who call themselves scientists. There are 92 elements (wrong); protons are indivisible (wrong again); one planet has rings (oops, two); amoebas are little animals and all bacteria are little plants (well, now they're all classified as Protista, neither animal nor plant kingdom)....

Things are not, of course, as bleak as that in all science classrooms. Some 20 years ago the Elementary Science Study (ESS) began producing a series of highly exploratory activities through which children could engage in real scientific exploration, discovery, theorizing, and verification, using safe, simple, inexpensive, and available materials. These curricula are still in print, and what I am about to propose involving language is

certainly no better. But to engage children in the process of figuring out how things work or what underlying patterns they follow, and especially to make that process feel natural and enjoyable, I think we must constantly look for subject matter that they can investigate without fancy equipment, advanced math, or vast sums of money and time.

Language is just such a subject and has the special advantage of already being part of the curriculum, not yet another addition to an already overcrowded schedule. Each of us has, in his or her own head, an enormous quantity of data about the way our language works. We have fairly ready access to this mass of data. We have even formed, albeit without being conscious of it, theories about language, and we have begun to do so before we ever set foot in school. The two year old is likely to utter phrases like *caught it* and *my feet* but abandon those forms two years later for words like *catched* and *foots* because these new versions (which the child has never heard an adult use) accord better with her theory of how language works. Only after a sufficiently large body of "exceptions" turns up do they warrant a new theory of their own, and the child then reverts back to the correct forms that she had once before used. Language is a perfect topic for scientific investigation.

My title also mentions play. Science and play are not so different for a young child. The language play of preschoolers is quite obvious, but it may not always be noticed that much of that play—including the often irritating repetitions of *our* language—constitutes a kind of experiment. This is science at its best, except for the lack of *articulation* of the rules that have been discovered.

That is where Logo comes in. A high-school senior who chose a programming course because she thought she should. She came to enjoy the computer much more than she expected, but was particularly impressed by her experiments with language. What moved her most was one discovery. As she put it: "I didn't know I knew so much!" She was referring to her knowledge—automatic and unthinking and, for that reason, discounted as simple and unimportant—of how to construct the plural of an English noun. Our computer did not know how to construct plurals, and so she had to teach it. The programming was simple enough for her to attend to the linguistic details, but how would she teach the computer explicit rules for what she herself did so automatically? It can be said, of course, that she already "knew" those rules in the sense that she could instantly generate any number of words and their plurals, but she was *aware* of only one rule, one that she thought she had learned in elementary school: "if the word ends in a *y*, change it to an *i* and add *es*." Like the once official line on protons, this rule is wrong, and she discovered it when she taught it to her program and it then faithfully converted *boy* into *boies*.

But why drag in the computer? Could her investigation not have proceded just as well without the computer? I think not. In part, it is difficult to test out ideas unless one has a slave stupid enough not to help. The computer is just that. Because my student was already fluent with her language and knew how to make the plural of the word *boy* she didn't think of it as "a word that ended with a *y*" but rather just as the word *boy*. Her discovery of *boies* was an accident. She was showing me her "finished" plural machine and typed the word *boy* in (perhaps because *boy* is not infrequently on the senior girl's mind?) and there it was: *boies*.

Another way that the computer helps is by allowing the problem to be dealt with in parts. A first cut at the rule might be represented this way:

```
TO PLURAL :NOUN
 OP WORD :NOUN "S
END
```

A next step acknowledges that words that end in *s* add *es* to make their plurals:

```
TO PLURAL :NOUN
 IF "S = LAST :NOUN [OP WORD :NOUN "ES]
 OP WORD :NOUN "S
END
```

Perhaps along the way, the student might decide to make a procedure that is a specialist on words ending with *y*.

```
TO YPLU :NOUN
 IF VOWELP LAST BL :NOUN [OP WORD :NOUN "S]
 OP WORD BL :NOUN "IES
END
```

The commands `PRINT YPLU "BOY` and `PRINT YPLU "LADY` will do the right thing, though `PRINT YPLU "MOUSE` will be amusingly wrong. Then the student might wish to incorporate this new specialist, a few more spelling situations and perhaps some exceptions into the original procedure.

```
TO PLURAL :NOUN
 IF MEMBERP :NOUN [SHEEP DEER] [OP :NOUN]
 IF :NOUN = "CHILD [OP "CHILDREN]
 IF MEMBERP LAST :NOUN [S O X] [OP WORD :NOUN "ES]
 IF "Y = LAST :NOUN [OP YPLU :NOUN]
 OP WORD :NOUN "S
END
```

The point is that a complete rule for making plurals can be quite complex. But any step along the way is simple to articulate and add to the description. If the description remained on paper, there would come a point when it was too complicated for people who are not linguistically trained and motivated to check. When the description is a computer program, the computer can do the checking.

Another exploration and articulation of what one already knows involves creating a program that takes, as inputs, the letters before and after the *ow* (as in *cow*) sound and then spells the word correctly, like this:

```
?PRINT SPELL "CL "N
CLOWN
?PRINT SPELL "CL "D
CLOUD
?PRINT SPELL "   "L
OWL
```

```
?PRINT SPELL "PL "
PLOW
?PRINT SPELL "PL "GH
PLOUGH
?PRINT SPELL "FL "ER
FLOWER
```

Again, nobody has been taught the rule, but when we see how *very* strange PLOWGH, COU, or FLOUER look, it seems like more than memorization of each individual word is at work. In fact, if you make up nonsense words with this sound in them, the chances are good that you will spell them "correctly"—that is, in accord with the rule that governs the vast majority of English words that contain this sound. A beginning for this procedure might be:

```
TO SPELL :HEAD :TAIL
 IF EMPTYP :TAIL [OP WORD :HEAD "OW]
 OP (WORD :HEAD "OU :TAIL)
END
```

What words does this get wrong? What *one* addition might you make to this rule to solve *one* of the patterns that it now gets wrong? What's another...?

Another intriguing notion is that you, who certainly do not know every word in the English language, have a very good notion of whether some unfamiliar combination of letters is perhaps a word you don't know or unlikely to be English at all. So what is the rule that governs what is acceptable as an English spelling?

This requires a different kind of exploration. One must learn how to make *source* procedures to supply needed chunks of language. Here are examples of sources for studying grammatical structure. They contain words and combinations of words.

```
TO WHO
 OP PICK [DANA DALE LEE SANDY]
END
```

```
TO DOESWHAT
 OP PICK [[SINGS TO] LOVES KISSES]
END
```

Here is another pair of sources that would be useful for beginning the study of how English words are constructed.

```
TO CONS1
 OP PICK [SH P ST M]
END
```

```
TO VOWEL
 OP PICK [A I O EA OI]
END
```

These procedures are so regular in their structure that they can be assembled as if from a cookbook. That is important because we want the programming to remain very much in the background. The only thing that differentiates these procedures is the language chunks that they contain. They all make use of the tool PICK that must be supplied for the student, preferably from a disk file so that the student does not even have to know that the procedure is not native to Logo.

```
TO PICK :OBJECT
 OP ITEM (1 + RANDOM COUNT :OBJECT) :OBJECT
END
```

Students must also learn how to assemble sources of component parts into structures built of those parts. Here are a few examples of such assemblies.

```
TO GOSSIP                              TO NEWWORD
 OP (SE WHO DOESWHAT WHO)               OP (WORD CONS1 VOWEL CONS1)
END                                    END
                TO QUESTION
                 OP (SE [WAS IT] WHO [WHO] DOESWHAT WHO [?])
                END
```

These procedures are also highly regular in structure and begin to suggest how the same sources can be used in different orders and combinations.

At the conference, I will present some patterns in adding prefixes and suffixes to Latin roots in English to investigate meaning, and in translating Italian or German to English.

Now, what else might a student investigate?

References

Birch, Alison. 1986. *The Logo Project Book*. Cambridge, Mass.: Terrapin.

Friendly, M. 1987. *Logo, A Language for Learning*. Hillsdale, N.J.: Erlbaum.

Goldenberg, E. P., and W. Feurzeig. 1987. *Exploring Language with Logo*. Cambridge, Mass.: The MIT Press.

Harvey, B. 1986. *Computer Science Logo Style, Volume 2: Projects, Styles, and Techniques*. Cambridge, Mass.: The MIT Press.

Harvey, B. 1987. *Computer Science Logo Style, Volume 3: Advanced Topics*. Cambridge, Mass.: The MIT Press.

Sharples, M. 1986. *Cognition, Computers, and Creative Writing*. Chichester, England: Ellis Horwood; Distributed in U. S. by J. Wiley.

© 1987 E. Paul Goldenberg

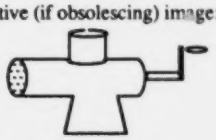
Making Functions Concrete

Explorations for Algebra and Pre-algebra

E. Paul Goldenberg, Ed.D.
Education Development Center, Inc.,
55 Chapel Street
Newton, MA 02160

The concept of functions is central in mathematics—so much so that the learner is constantly brushing up against it, even without knowing it—yet it is seldom well learned. It is often considered abstract and difficult to teach. This paper suggests that appropriate visual imagery and the opportunity to make functions concretely manipulable by giving them life on the computer makes functions become more accessible and interesting.

So I will begin with a suggestive (if obsolescing) image:



When we drop a chunk of beef in the hopper and turn the crank, we expect ground beef and not applesauce to drop out the spout. For a particular input, we have very strong expectations about the output. Functions are like meatgrinders, but in mathematics, we are seldom satisfied with "strong expectations." We demand certainty, proof. So think of a function as a kind of meatgrinder with an ironclad guarantee about what comes out:

> The machine will not necessarily accept everything that you might think of dropping into it. It can sometimes clog or spit up. But *if* something comes out for a particular input, then the very same something will come out the next time you give it the same input. Whatever comes out is completely determined by what went in. (You may not know what to expect will come out when you throw a banana split in, but whatever comes out *this* time will also come out the next time you throw in a banana split. No randomness, no jokes, and no exceptions allowed.)

We do not think of functions as "grinding" their inputs. We may regard them as transforming the input into the output or perhaps as matching inputs with outputs. Below are several examples of functions. Each is represented as a table.

A		B		C		D		E		F	
word	→ words	creation	→ crtn	0	→ 0	0	→ 0	0	→ 0	0	→ ?
cat	cats	carton	crtn	1	1	1	1	1	1	1	1.0
box	boxes	certain	crtn	2	4	2	0	2	10	2	0.5
octopus	octopuses	cartoon	crtn	3	9	3	1	3	11	3	0.3
dog	dogs	captured	cptrd	4	16	4	0	4	100	4	0.3
sheep	sheep	eerier	rr	5	25	5	1	5	101	5	0.2
child	children	child	chld	6	36	6	0	6	110	6	0.2

We often assume we are working with functions even when we do not acknowledge them explicitly. For a simple example, imagine a student being shown the pattern in column B and being asked "What would come out if I put in 'season?'" In its wording, the question seems to imply that there is *one* correct answer and that it would be just as true tomorrow as it is today. As can be seen in the first entry in column F, functions do not require that there always be an answer to the question "What comes out if I put in ___?" Neither do they require that there be a different answer for each question, as can be seen in columns B, D, and F. But if there is an answer, there'd better be only one.

Functions can not only transform words to words or numbers to numbers as in the examples given above, but can also transform words to numbers (taking in, for example, a student's name and outputting the student's height) or numbers to words (taking in a number on a football jersey and outputting the bearer's name). In fact, any transformation that respects the one answer rule is a function.

Any one function may be represented in many ways. Above were six functions represented as tables. Here are the same six functions represented as English statements.

Output the plural of the input word.
Strip the letters a, e, i, o, and u out of the input word and output the remaining letters.
Output the square of the input number.
Divide the number by 2 and output the remainder.
Output the binary representation of the input number.
Round the reciprocal of the input to one decimal place and output that result.

Here is yet another way of representing the functions.

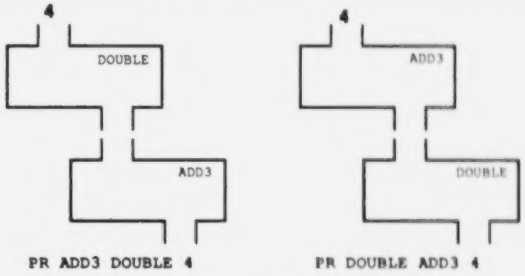
```
TO AY :INPUT
 IF MEMBERP :INPUT [SHEEP DEER] [OP :INPUT]
 IF :INPUT = "CHILD [OP "CHILDREN]
 IF MEMBERP LAST :INPUT [S O X] [OP WORD :INPUT "ES]
 OP WORD :INPUT "S
END

TO BEE :INPUT
 IF EMPTYP :INPUT [OP "]
 IF VOWELP FIRST :INPUT [OP BEE BF :INPUT]
 OP WORD FIRST :INPUT BEE BF :INPUT
END

TO CEE :INPUT
 OP :INPUT * :INPUT
END

TO [illegible]
 [illegible]REMAINDER :INPUT 2
END

TO EFF :INPUT
 OP (ROUND 10 / :INPUT) / 10
END

TO EEE :INPUT
 IF :INPUT < 0 [OP (0 - EEE (0 - :INPUT))]
 IF :INPUT < 2 [OP :INPUT]
 OP WORD (EEE INT (:INPUT / 2)) (DEE :INPUT)
END
```

When the functions are numerical, they may often be represented sensibly as graphs. Tools to help students examine graphic representations of arbitrary numerical functions will be presented later in this paper.

Because Logo makes the programming of many numerical functions (like, for example, CEE, DEE, and EFF) particularly easy, students—even those who have only a small amount of Logo knowledge—can write their own functions and explore their behaviors without having their attention diverted from mathematics to the nuts and bolts of computers and programming. By writing the functions themselves and exploring the behaviors on the computer, what might otherwise be an abstract paper-and-pencil exercise made unpleasant by large amounts of computation becomes amenable to easy experimentation.

Through two such explorations, this paper shows how the presentation—abstraction vs. concrete task—affects the accessibility of the explorations. Explorations that may otherwise have seemed suitable only for older students with advanced mathematical backgrounds may become fully accessible to young elementary students when they are presented with suitable imagery and suggestions for concrete action.

Composition of functions: When is f(g(x)) = g(f(x))? Posed in this way, with vocabulary and mathematical notation that are likely to be unfamiliar to many students even in secondary (and some postsecondary) courses, the questions remain abstract even after the terms and symbols are defined. Here is an image, some Logo, and an alternate wording that suggests concrete actions to take. (My choice of wording here was to make the task clear to you, an adult reader. Appropriate presentation for elementary school children might initially be through discussion and example rather than a printed paragraph.)



The machine on the left shows a 4 going through a processor that doubles it and then the result going through a processor that adds 3. What comes out the bottom? The machine on the right shows a 4 going through a processor that adds 3; whatever comes out then goes through a processor that doubles that result. What is the final result this time? So, the two machines produce different results when a 4 is put into them. But perhaps they might give the *same* result if a 5 or a 0 or a 17 (or a -4 or a 2.3) were put in. Find an input number that *does* make the results come out the same or give some reason to believe that there is no such number.

Here are the Logo definitions of each of the processors.

```
TO DOUBLE :X
 OP 2 * :X
END
```

```
TO ADD3 :X
 OP :X + 3
END
```

Let's consider how students at different levels might approach the problem.

Algebra students might be encouraged to express each machine's function in terms of the composition of its processors. So the left hand machine is a $2x + 3$ machine whereas the right hand machine is a $2(x + 3)$ machine. Merely representing these concrete objects algebraically is a worthwhile exercise for the beginning algebra student, requiring understanding of the use of parentheses and suggesting a useful image for the distributive property of multiplication. The question then becomes, for what values of x is $2(x + 3)$ equal to $2x + 3$? To represent this algebraically, the student may care to write the "equation" $2(x + 3) = 2x + 3$ and solve for x. Since it is a common algebra manipulation error to remove parentheses without multiplying through, some students may feel that the algebra proves that the statement is true for all values of x. Experimenting with various values as inputs for the machines, however, willnot reveal any values of x for which this statement is true. But the inability to find any is no proof that there are some that we simply have not yet tried. When parentheses are removed correctly, the statement becomes $2x + 6 = 2x + 3$, which can be true only if there is some value $2x$ which can have either a 6 or a 3 added to it to produce the same result. Clearly silly.

The problem might well be extended in the following way.

> Find a pair of processors that *do* produce the same results for a given input regardless of the order in which they are used? Does your pair produce the same results for*all* inputs, or does it work only for some but *not* all inputs? Whichever kind you found, can you find the other kind?

To make the wording less complex and offer a bit more support for a beginning student, the extension might be posed with even more concrete suggestions for action.

> Here are 6 processors: DOUBLE, TRIPLE, ADD3, ADD5, SQUARE, CUBE. You've already tried one pair of them and found that DOUBLE ADD3 is not the same as ADD3 DOUBLE. Are there any pairs of these processors for which the order does *not* matter?
>
> ```
> TO SQUARE :X
>  OP :X * :X          etc.
> END
> ```

Pre-algebra students lack the formal tools to express the problem algebraically and so must explore the problem numerically or graphically. Here is one sixth grader's argument that DOUBLE ADD3 is never going to be the same as ADD3 DOUBLE. Her argument rests on numerical regularity.

> I constructed a table like this:
>
:x	DOUBLE ADD3 :x	ADD3 DOUBLE :x
1	8	5
2	10	7
3	12	9
4	14	11
5	16	13
>
> DOUBLE ADD3 :x is always 3 bigger than ADD3 DOUBLE :x, and it is always going to be! So they can never be equal.

The student's assertion "and it is always going to be" is not satisfactory in formal proof, but her assumption that the functions she is dealing with are "well behaved" is quite reasonable. It is a good argument.

Provided with a suitable set of graphing tools on disk, students may represent and examine functions graphically. The tools include WIDEANGLE or CLOSEUP to enable students to choose a scale on which to draw their graphs. The commands GRAPH [DOUBLE ADD3 :X] and GRAPH [ADD3 DOUBLE :X] will show that these two functions are never equal: they never cross. Similarly, students might find the solution to $2(x + 3) = 3x + 5$ by drawing a CLOSEUP of both GRAPH [DOUBLE ADD3 :X] and GRAPH [ADD5 TRIPLE :X] and using SETPOINT to examine the intersection—the place where $2(x + 3)$ and $3x + 5$ are equal. SETPOINT takes an *x*-coordinate and a *y*-coordinate and moves the turtle to the specified point (at the current scale) SETPOINT 1 8 will land the turtle on the intersection of $2(x + 3)$ and $3x + 5$. The grapher allows examination of all sorts of functions as long as the variable is named :X, so the command GRAPH [(20 * SIN (8 * :X)) + (0.01 * SQUARE (:X + 20)) - 30] is legal. (But CEE, which is otherwise equivalent to SQUARE, cannot be used with GRAPH.)

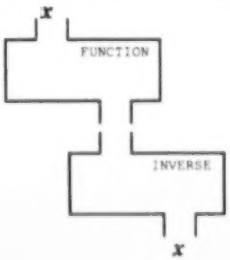
```
TO CLOSEUP
 MAKE "SCALE 10
 AXES
END

TO AXES
 CS
 HT
 REPEAT 4 [AXIS RT 10]
 ST
END
```

```
TO WIDEANGLE
 MAKE "SCALE 1
 AXES
END

TO AXIS
 REPEAT 10 [FD 10 BLIP]
END
```

```
TO BLIP
 RT 90
 FD 2 BK 4 FD 2
 LT 90
END
```

```
TO GRAPH :FUNCTION
 IF NOT NAMEP "SCALE [WIDEANGLE]
 LOCAL "X
 MAKE "X -120 / :SCALE
 WINDOW
 PU
 SETPOINT :X RUN :FUNCTION
 PD
 PRINT (SE :FUNCTION [FROM] :X [TO] (0 - :X))
 GRAPH1 :FUNCTION :X (0 - :X) 1
END

TO GRAPH1 :FUNCTION :X :XMAX :INC
 IF :X > :XMAX [STOP]
 SETPOINT :X RUN :FUNCTION
 GRAPH1 :FUNCTION :X + :INC :XMAX :INC
END

TO SETPOINT :X :Y
 SETPOS LIST (:SCALE * :X) (:SCALE * :Y)
END
```

Inverses of functions: When is f(g(x)) = x? Here is an image.



Students encounter variations on this theme all the time, again mostly without recognizing them. One such variations is suggested by the following table that a student in elementary school might be asked to complete after discovering the pattern.

2	5
4	9
7	
3	7
10	21
0	
	3

In this case, the question builds in the assumption that the function has an inverse: that one can figure out an entry in either column from the corresponding entry in the other column. That is the case for many functions, but is not the case for many others. For example, it is not the case in functions BEE, DEE, or EFF above.

The search for the inverse is also subtly contained in the manipulations an algebra student is asked to perform when solving for x in a statement like $3x + 5 = 17$. The problem is equivalent to asking what input would have to be put into the processor FUNCTION (defined below) to get it to output 17.

```
TO FUNCTION :X
 OP 3 * :X + 5
END
```

But algebra students are not generally encouraged to solve problems of this kind by trial and error or by examining the graphs. Instead, they are encouraged to subtract 5 from the 17 and then divide the result by 3. Generalizing the idea, one might code that in Logo as:

```
TO INVERSE :X
 OP (:X - 5) / 3
END
```

The statement `PRINT INVERSE 17` will give the answer to the original problem, but there is richness to be gained by not dropping the problem immediately. Recall the earlier problem of finding a pair of functions that might be applied in either order to yield the same result. Is this such a pair? Will the machine `FUNCTION INVERSE FUNCTION` invert `INVERSE INVERSE FUNCTION`? What about `FUNCTION INVERSE INVERSE FUNCTION`? Is there anything interesting to say about the graphs of various functions and their inverses?

Principles: Perhaps the most important pedagogical principle I am trying to convey here is the idea of using computing to make the abstract more concrete. To do that, the programming must remain simple enough, repetitive enough, formulaic enough never to attract much attention. Students may be taught simple rules like "read from the bottom up" for translating the machine diagrams into programs. They may be taught to think of what process a number goes through first, second, third, etc. They learn the "formula" for a Logo function:

```
TO WHATEVERFUNCTION :X
 OP some mathematical expression involving :X
END
```

Programming must not just be kept out of the way; it must be used. So students use it to build runnable models of ideas they generate in other ways. Some of these ideas are generated using a consistent imagery of processors with hoppers and spouts assembled into machines. Students learn that machines may, themselves, be treated as single processors and may be assembled with other machines to make yet more complex machines. Students learn to translate from one representation (table, English definition, algebra, machine diagram, Logo) to another.

Finally, students use the machines both for practical purposes like solving or examining a problem and also for what might be called theoretical purposes. They begin to examine the properties of the machines and assemblages of machines themselves. This transition, like the translation among different representations of functions, helps to teach the process of abstraction that mathematics is all about.

There is a reason for caring about all of this, and a reason why I pitch this paper toward the introduction *and* follow-up of the study of elementary functions.

> *[Twelfth grade] achievement in algebra and elementary functions...for fifteen countries shows the U.S. to score among the lowest fourth of participating countries. Much scientific and technical work, in college and elsewhere, requires competence in these basic areas of mathematics.* (From a national report on the second international mathematics study funded by the NSF, NIE, and NCES. Curtis McKnight, et al., 1987. *The Underachieving Curriculum: Assessing U.S. school mathematics from an international perspective.* Champaign: Stipes Publishing Co. p. 22.)

THE TURTLE GROWS UP: LOGO Applications in High School Physics, English, Biology and Artificial Intelligence

Janet Bard Hanson
University of South Florida

To provide a background for the presentation of applications of Logo in the high school setting, think with me for a few minutes about some important metaphors. In the industrial age, machines were built to extend the muscles of man, to accomplish work through levers, and other tools. In the more recent post-industrial age, computers have been developed to extend the mind of man. However many uses of computers, especially in the educational arena, continue in the model of the machine of the industrial age. The data processing model sees input as the child coming into the school situation (raw materiels), processing as what happens in the school, and the output as the student, citizen, potential worker. The "factory" model, similiar to Henry Ford's assembly line, places an emphasis on efficiency (less resources, faster time), uniformity (national curriculum, texts), quality control (passing assessment tests), and productivity without regard to individual learning styles, creativity, or problem solving. The majority of the current computer assisted instruction (CAI) in use today models this mechanical, repetitous, assembly line mode. Studies of conventional CAI have confirmed that children learning via computer acquired basic skills faster, learned more, and retained better than children learning by more conventional means.

The new metaphor pictures the computer as an extension of the human mind. Research in artificial intelligence has provided a richer meaning of the mind and development of intelligence. A child's education must go beyond the acquisition of stable facts to higher thought processes; analysis, synthesis, and evaluation. Herbert Simon defines intelligence as the ability to solve problems in an ever changing environment. Papert defines intelligence as the ability of the organism to adapt to the environment. David Wechsler sees adaptation of behavior as the key. The chief task of education would appear to be the development of a thinking, adapting, political adult able to deal with a great number of variables and changes in his environment as he/she works to develop a better world.

Logo can serve as the key in nurturing this intellectual growth. Logo enables educators to capture all the marvelous capabilities of the computer to provide the fertile environment of exploration and manipulation of variables. This is of chief importance during the high school years. At this advanced stage in development, students have the cognitive capabilities inherent in the stage of formal operations which allows them to imagine an infinite variety of possibilities and develop hypothetical reasoning. Piaget has written that the adolescent possesses a large number of cognitive operations to attack problems. The learner can deal with a problem in numerous ways from a variety of perspectives. Computer microworlds can be developed in most curricular areas. Through the manipulation of variables and problem solving strategies, the student can discover new relationships and exercise intellectual skills to the fullest.

In contrast to expository teaching, the active world of list processing and advanced turtle graphics using embedded recursion can give students many experiences with content and subject areas. Concepts can be demonstrated in concrete ways, turtle graphics can dynamically illustrate concepts and students again can change parameters to discover laws and relationships. In manipulating the content, the student learns facts in an efficient and meaningful way.

A first example is high school English. In the curricular areas of literature, grammar, and composition, students become actively involved while learning. In the study of literary works, the characters and their interrelationships often become cloudy and confusing to students. A typical Logo exercise first asks students to assign the various characters different size circles according to their importance in the development of the work. A minor character would be given a small circle, a major character a larger circle. The student would have a graphic illustration of the characters and their importance. The next step is determining the interrelationships between the various characters. The student would place the character circles around the screen with placement and positon according to the plot. The graphics can be highlighted with color and different screens can be developed for parts of the work. The students can compare and contrast their graphics leading to lively discussions involving various interpretations.

In grammar, the structure of language and syntax rules can be illustrated by diagramming sentences. This is easily translated into Logo lists allowing for manipulation and discovery of relationships. In composition, new ideas for development of plots can be stimulated by reviewing the concept of recursion. The popular television series Twilight Zone is replete with recusive themes.

In the study of Physics, many opportunities arise for the incorporation of Logo. In addition to the Dynaturtle exercises for illustrating force and the principles of mechanics, many structures and relationships can easily be developed using turtle graphics. Fractals can be formed by using embedded recursive techniques. The Sierpinski gasket consists of triangles within triangles continuing to the finest level.

The study of math have many areas for modeling with Logo. Statistics and numerical analysis are more understandable with examples finding combinations and learning counting principles. Rotation of geometric figures is an interesting challenge for trigonometry students.

The concepts of Artificial Intelligence can be illustrated easily using Logo. The basic concepts of an expert system are developed through tree structures. With list processing and turtle graphics, knowledge trees can be built. Simple programs can simulate breadth and depth searches using concepts of queue or stack. Principles of robotics can be illustrated using the "educated" Logo turtle. Natural language comprehension can be understood with rules of syntax and sentence structure.

The possibilities seem endless and spill over into all areas of science. Taxonomy in Biology can be understood with tree structures and list processing. The hierarchical relationships and organizations structures in Government and Business can be mapped. The concepts of associative lists and properties are used to aid in the understanding of chemical elements and the Periodic Chart in Chemistry.

The activities presented combine a variety of levels of programming sophistication. Many examples contain preprogrammed procedures needing little prior knowledge of Logo while others require an understanding of recursive concepts. As students develop new procedures, the body of examples can grow as new Logo vocabulary develops.

The computer and the capabilities of Logo to manipulate variables, create new vocabulary, and build hierarchical lists, combine to offer fertile ground for the extensive of the mind. As students develop models and experiment with the various parameters, relationships are discovered and reinforced. Within these microworlds, students explore strategies for problem solving, analyze elements and interrelationships, synthesize abstract relationships, and evaluate criteria to form judgments.

"Logo LinX With Traditional K-8 Curricula"

Judi Harris

Curry School of Education, University of Virginia
Educational Studies Department
405 Emmett Street, 287 Ruffner Hall
Charlottesville, VA 22903

Mozart used billiard balls. Descartes watched a fly on the ceiling. Papert played with gears. A friend of mine takes a shower. I browse in bookstores. Genius or not, we all seem to have a favorite method that helps inspire creative ideas.

Whatever the method, we all must "get the creative juices flowing" to develop more ways to integrate Logo explorations into traditional subject study. The purposes of this session are to present both a potpourri of Logo applications ideas across elementary and middle school curricula, and suggested methodologies for the development of such activities.

These efforts are necessary for reasons that have become apparent during the last few years of widespread Logo use. Teachers and administrators are beginning to question using many school hours solely for turtle graphics work. Inadequate in-service training can blind practicioners to the endless possibilities of interdisciplinary Logo work, or frustrate those who want to learn more, but have few resources to consult for assistance. Renewed pressure for "excellence in education" is operationalized in many districts by the issue of long lists of curricular goals and objectives. These may seem to preclude the "extra time" perceived necessary for Logo work by classroom teachers already stressed by scheduling demands from administrators, parents, and school specialists.

The integration of Logo exploration into traditional subject study can, at least, be viewed as a time-saver. Using Logo to help students learn about latitude and longitude, for example, concurrently accomplishes both map skills and computer literacy objectives quite commonly required at upper elementary levels. It may also allow more time each week to be spent on Logo work than if map skills and computer literacy activities are kept separate.

More importantly, though, interdisciplinary Logo explorations closely parallel the natural functioning of the child's mind. Children don't compartmentalize inquiry into language arts, math., social studies, science, art, music and gym; rather, their minds cut across the boundaries between subjects that we adults errect to "organize" their instruction. Specific curricular goals can quite successfully be integrated into larger areas of study. These can, in turn, be organized in ways that reflect children's cognitive styles, rather than an adult's mental model of subject content. Logo can serve as a flexible, facilitative medium in this endeavor.

Back to the bookstore. As I browse, I scan all of the sections with no particular subject in mind. I am looking for Logolike patterns in other areas of interest. Soon, books of ideas make themselves apparent:

--angles formed by body positions in photographs of gymnasts, skiers, and dancers;
--nonsense words formed from meaningful syllables, and defined humorously;
--pictures composed entirely from circle sections.

These potential Logo challenges catalyze other thoughts:

--learning dance steps through Logo commands;
--learning to spell by rules, rather than memorization;
--defining circles by radius, rather than circumference.

The process continues as I point out possible projects to the students, emphasizing pattern similarities among different subject areas. Their thoughts inspire more ideas for all of us, as does their work.

Each of us has our own way to open our minds to new connections between familiar topics. The important component is not the place or the stimuli; books in a bookstore are no "better" inspiration than warm water in the shower. Rather, providing for the **opportunity** for such creative brainstorming to occur is crucial. As seeds germinate best in moist environments, so do powerful interdisciplinary Logo project ideas spring from a rich cognitive environment. From the teacher's vantage point, this necessitates

- --a thorough understanding of Logo primitive functions and a good variety of procedure structures at hand, in both graphics and text domains,
- --a thorough understanding of the content and processes inherent in the body of knowledge (curricular area) to be explored,
- --and a willingness to draw new parallels and make new connections between the two.

I write a monthly column in Logo Exchange magazine ("**Logo LinX**,") in which I describe some of the interesting interdisciplinary connections that I have explored with my students through Logo use or Logolike activities. I plan to demonstrate and discuss many specific project ideas that bridge Logo exploration with more traditional subject study, such as

- --language arts,
- --social studies,
- --mathematics,
- --science,
- --music,
- --art,
- --and physical education.

Some of these projects are listed on the following page. I hope that my ideas will help to inspire yours.

Logo LinX to Traditional Curricula

Language Arts

Structural Skills (spelling by rules,)* "Sniglets,"* word and sentence shapes,** palindromes, language translators, cursive writing, brand name challenges, a "Logolike quotes" database for teachers

Social Studies

Hieroglyphic typewriters,* cooperative neighborhood construction, subway systems and topology,** Indian dwelling shapes that reflect culture and environment, chanting patterns in religious texts,* latitude and longitude tools, symbol system tools

Mathematics

Angle and length estimation diagnostic assessment tools,** "hairy polygons" and the Euclidean toolbox,** digital time patterns,* Ed Emberley drawing projects, graphic representations of number patterns, a pattern block microworld, popular game microworlds (such as dominoes and "Othello")**

Science

3D polyhedra construction,* constellation mapping,** snowflake crystallization patterns, simple machine diagrams, molecule diagrams, leaf vein patterns, "invent-an-animal," forest construction (branching structures)

Music, Art and Physical Education

Fractions and musical notation,* historical trends in musical form,** "Droodles,"* Logo choreography,** body angles,* pointillism

* indicates the subject of a 1986-87 "Logo LinX" column
** indicates the subject of a "Logo LinX" column planned for 1987-88

Reprints are available upon request.

Tool Procedures for Efficient Development of Logo-Based Applications: Text Screen Formatting Tools

Francis A. Harvey

Leadership, Instruction, and Technology Department
College of Education
Lehigh University
Bethlehem, Pennsylvania

Logo for Logo's Sake and Logo for Applications

Logo's Original Goals

Logo was originally developed as an environment for introducing novices, particularly children, to the world of computers and computer programming. The widespread adaptation of Logo in schools has demonstrated that Logo has admirably fulfilled this original expectation.

Original uses of Logo, both because of computer memory limitations and because of lack of awareness of Logo's great potential for other uses, were for the most part limited to the use for which it was originally intended: Logo was used to "study Logo."

Logo as an Applications Language

Many educators who were first introduced to computers and computer programming through this "Logo for its own sake" approach soon recognized the potential of Logo as an environment for producing powerful and useful computer applications for instruction as well as for other uses. They were frustrated, however, by the limited capabilities of early versions of Logo and the microcomputer hardware on which it ran.

The first version of Logo available for microcomputers was developed for an Apple II microcomputer with at least 64K of RAM memory. That version provided the user with approximately 2200 nodes of workspace, completely adequate for turtle graphics explorations, but hardly enough room to develop a meaningful application.

The Advent of Better Logos and More Powerful Microcomputers

The availability of microcomputers with extended RAM memory, 256K or more, and the development of new and more powerful implementations of Logo for these machines (for example, a compiled version of Logo for the Apple Macintosh), provides a much more adequate environment for the development of extended applications. It is expected that applications developers will soon begin to take advantage of this environment, and the number of applications written in Logo will increase substantially in the future.

Developing Screen Text Displays

Much of the programming effort involved in developing a computer application, whether the application is an instructional application or of some other type, and whether the development takes place in Logo or in another programming environment, consists of generating text screen materials. Text displays are used for the display of titles,

instructions, menus, questions and other user prompting messages, tutorial information, and other purposes.

This paper will describe a set of tool procedures developed to facilitate text screen formatting and make the overall task of applications development easier and more efficient. Appendix A includes a sample listing and sample output from one of the procedures to be described, to ullustrate the use o these tool procedures.

Procedures for Formatting Text Screens

The standard process for developing text screens involves writing procedures which use a combination of cursor positioning primitives (SETCURSOR in LCSI Logo versions and CURSOR in MIT versions) and PRINT to display each line of text. Development usually proceeds by trial and error, and can be both tedious and time-consuming. Since much of the work involved in this process is straightforward and repetitive, it is the kind of work for which the computer is well suited. The following procedures shift much of the work of formatting text screen displays from the computer programmer to the computer itself, freeing the programmer to concentrate on larger issues.

Word Wrap

Using the standard method, text which is longer than the number of columns of screen display must be broken down into separate lists. Otherwise words may be split at the wrong place at the end of a line, making the text difficult if not impossible to read. The WORD.WRAP procedure takes a list of any length as input and prints the list on the screen. Before printing each word in the list, however, the procedures checks to see whether there is room on the current line for that word. If the word will not fit, the procedure issues a carriage return to move the cursor to the beginning of the next line before printing the word. The procedure then recursively calls itself with the remainder of the list as input.

WORD.WRAP takes as inputs the list of text to be displayed, the number of columns of display desired, and a dummy variable (:POINTER) which indicates current column position. Thus, text of any width can be generated, and the width of text can be varied from paragraph to paragraph on the screen..

Word Wrap (Multi-Space)

WORD.WRAP.SPACE is an extension of the WORD.WRAP procedure described above. WORD.WRAP.SPACE takes as an additional input the number of lines of spacing desired for the text display. The procedure issues that number of carriage returns at the end of each line, thereby producing double- or triple- or whatever-spaced text. Double-spaced text is usually easier to read, especially if the text in in all upper case. As with WORD.WRAP, the line length is also an input.

Centering Text

CENTER.TEXT takes a list or word as input. The procedure then counts the number of characters in the input and computes the appropriate column in which to begin displaying the text so it will be displayed centered on the scren. The procedure then moves the cursor to that position and prints the text. If the length of text to be displayed is larger

than the width of the screen, the text is broken down into segments, each of which is displayed as centered text on separate lines.

Underlining Text

Since most microcomputers are not capable of displaying underlined characters, the computer must be "fooled" into producing a display which resembles underlined text. The UNDERLINE procedure takes a list or word as input. The procedure then generates a list of dashes (----) which matches the input word or list. The procedure then prints the input word or list on one line and prints the dashes on the line just below, so that a dash appears under each character (but not under spaces between characters).

Centering and Underlining Text

CENTER.AND.UNDERLINE takes a word or list as input, and uses the procedures described above to produce underlined text centered on the screen.

Centering Text with a Border

CENTER.IN.BOX takes a word or list as input, and displays this input centered on the screen, surrounded on all four sides by a border of asterisks or other text character selected by the user. For example, the procedure call

CENTER.IN.BOX [Instructions: Part One] would produce the display shown in Figure 1.

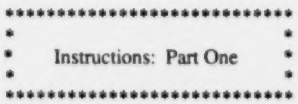


Figure 1. Sample Output from the CENTER.IN.BOX Procedure

Summary and Conclusions

The text formatting procedures described in this paper were developed for use as tools by developers of Logo applications. They are designed for use by novice programmers as well as experienced applications developers. Their purpose is to shift some of the routine and repetitive work of developing applications from the user to the computer, so that the human programmer can concentrate his or her energies on more complex issues.

These procedures also provide interesting examples of the application of Logo's list processing capabilities. As such they are useful objects of study in themselves.

It is hoped that other Logo users will find the procedures described in this paper useful for their intended purpose as tools for applications development. In this way the procedures will contribute to the development of Logo as a full-fledged applications language.

Appendix A

Procedures for Text Formatting: Sample Procedure Listing and Sample Output

Space does not permit the inclusion of listings of all procedures described in the paper. The listing for the procedure WORD.WRAP.SPACE (written in IBM Logo) is presented here as an example. A set of complete procedure listings is available from the author.

```
TO WORD.WRAP.SPACE  :TEXT  :TEXT.COLUMNS  :LINE.SPACE  :POINTER
   IF :TEXT = [ ]  [PR "  STOP ]
   MAKE "POINTER :POINTER  +  1  +  COUNT FIRST :TEXT
         IF :POINTER  <  ( :TEXT.COLS  - 1 )                    } Type
            [ TYPE FIRST  :TEXT  TYPE  "   WORD.WRAP.SPACE      } this
            BF  :TEXT  :TEXT.COLS  :LINE.SPACE  :POINTER]       } as
                                                                } one
            [REPEAT  :LINE.SPACE  [PR " ]  WORD.WRAP.SPACE      } long
             :TEXT  :TEXT.COLS  :LINE.SPACE  0]                 } line
END
```

To create a sample text passage to demonstrate this procedure, enter:

```
MAKE "SAMPLE.TEXT [You can also see ships stopping at sugar refineries along the
Delaware River.  The sugar refineries are not as large as oil refineries.  But they are located
there for the same reasons as the oil refineries.  They can get raw sugar cheaply by ship.
There is also a good market for sugar in the nearby cities. ]
```

The instruction PRINT :SAMPLE.TEXT would produce (on an 80-column screen) the following output, with words split every 80 characters:

```
You can also see ships stopping at sugar refineries along the Delaware River.  T
he sugar refineries are not as large  as oil  refineries.  But   they are located th
ere for the same reasons as the oil refineries.  They can get raw sugar cheaply
by ship.  There is also a good market for sugar in the nearby cities.
```

WORD.WRAP.SPACE :SAMPLE.TEXT 55 2 0 would automatically display the paragraph double-spaced, on lines 55 columns wide, without breaking up words:

```
You can also see ships stopping at sugar refineries

along the Delaware River.  The sugar refineries are

not as large as oil refineries.  But they are located

there for the same reasons as the oil refineries.

They can get raw sugar cheaply by ship.  There is

also a good market for sugar in the nearby cities.
```

COPYRIGHT 1987 Gayle Lawrence

Logo: Where Models and Reality Connect

by Gayle Lawrence
Brittonkill Central School

On a recent test given by the National Assessment of Educational Progress, students were asked to solve this problem: John and Lewis are planning a rectangular garden 10 feet long and 6 feet wide, and they want to put a fence around it. The question, simply put, was: "How much fence do you need?" Of the 13 year olds taking the test, 31 percent said 32 feet; 38 percent, 16 feet; and 21 percent, 60 feet, with 10 percent giving other answers. (Hechinger, 1986) Most of the students relied solely upon their memory of the perimeter for a rectangle formula to solve the problem. Probably, it did not occur to them to sketch or model the garden plot to arrive at the correct answer. Their experiences had not provided them with the appropriate model of a rectangular figure nor had they developed a modelling approach to problem solution.

Examination of approaches to teaching mathematics suggests that these results should come as no surprise. Although teachers affirm the value of intuitive approaches and the use of concrete models and other aids to promote spatial visualization and understanding, they report that they mainly use formal, "tell and show" approaches to instruction in geometry. Integers are often introduced using the number line and other physical examples. However, when operations using integers are introduced, teachers report reliance on giving rules. (Travers and McKnight, 1985)

Over a decade ago, curriculum reformers called for process oriented curricula to replace information based curricula. (Burns and Brooks, 1970) Between 1975 and 1980, public 4-year colleges increased their offerings in remedial mathematics courses by 72 percent. (National Commission On Excellence in Education, 1983) Clearly, there is a problem with our current approach to mathematics education. More of the same is not the answer. We need to provide students with experiences in the construction of models of reality and to enhance their ability to make the important connection between their models and real problems.

Formally, a model is an abstract description of the real world; it is a simple representation of more complex forms, processes, and functions of physical phenomena or ideas. A model is constructed to facilitate understanding and enhance prediction. (Rubenstein, 1975) It is a purposeful and often radical abstraction. It should contain only the elements of reality needed to solve the problem. (Judson, 1980)

The use of verbal, diagrammatical and mathematical models is fundamental to human thinking. Children demonstrate extraordinary ability to model reality as they perceive it. They use clay, sand, blocks, paper, almost any manipulative available, to build and rebuild models of the world around them. They model objects, behaviors, language, and constantly experiment with materials, ideas and actions as a means of understanding and interpreting the environment. Modelling possesses a special quality of being at once playful and serious. Whether the modeller is an adult or a child, modelling allows them to "play around" with serious ideas as a means of expanding and creating knowledge of how things are or might be. (Judson, 1980)

Great modellers are able to extract the essence from often confusing and irrelevant information and to perceive a means of modelling a phenomena or process in a unique way. Albert Einstein's brilliance was in the formulation that linked previously unconnected notions of space and time in a comparatively simple mathematical model, which we now know as

special relativity. His ability to pose the right question, and create the right model, resulted in a tremendous contribution to our understanding of the universe. (Cross and Moscardini, 1985) To further appreciate the importance of models to society's advancement, think of the significance of Newton's model to explain gravitation, Maxwell and Clerk's model of electromagnetic waves, Watson and Crick's model of the DNA double helixes, Leonardo da Vinci's models of flying machines, and Lawrence Klein's models of the American economy.

One of the most exciting and inspiring examples of modelling involves Linus Pauling's discovery of the alpha helix protein structure. A close look at his experience deepens one's appreciation for the blend of art and science involved in the modelling of complex phenomena. Pauling had studied the chemical and structural properities of protein chains for more than ten years. Many chemists of the time were studying these as well. The "aha!" came to Pauling as he lay in bed recovering from a cold. He sketched precisely the skeleton of a possible basic structure for protein chains. His theory began as a zig-zag-zig sketch on paper, which he then folded into a tube to achieve the three-dimensional effect required to represent the structure. Pauling's genius was in his ability to perceive the essence from those facts and relationships and to model what was going on. He extracted the quintessence from the jumble of knowledge and made a substantial contribution to the advancement of scientific understanding of protein structures with his model. (Judson, 1980)

Everyday life is replete with examples of useful models: a doctor's surgical notes, a teacher's lesson plan, a carpenter's sketch, a seamstress's pattern, a cartographer's map, an artist's painting, and, as Plato pointed out in 400 BC, almost every noun we use. (Cross et al., 1985) Reflection upon the countless ways in which we are involved, directly and indirectly, with models makes a strong case for improving modelling skills in our students.

Let us not pretend, however, that one can learn to model just by examining examples of models. Lecturing or "talk and chalk" methods which rely on recall and fact accumulation will not teach someone how to model. Modelling is an art that draws upon knowledge and experience. A metaphor used by J.H. Mason (1984) suggests that learning to model is very like becoming a potter. We cannot promote children's modelling skills by simply talking to them about models or showing them models any more than we could enable them to create pottery by showing them pots.

Modelling is an interacitve process; it requires involvement, participation, enquiry and exposure. Effective modelling requires discrimination, critical assessment, hypothesis formulation and testing, generalization, analysis, synthesis, evaluation, creativity and intuition. It is difficult to learn, impossible to teach in a "rote-memory" based environment, and, indeed, very powerful.

In my experience as an elementary computer resource teacher in a computer lab setting, I have come to the realization that the Logo based learning environment is especially suited to promoting modelling skills in young students. As we worked on our polygon exploration, I saw that we had the ideal laboratory for learning about models and how they connect to the real world. I could see the power and importance of modelling as a problem solving approach, and I began to focus my attention primarily upon the process of modelling, and secondarily, upon the objects being modelled. The Logo lab provided an opportunity for exercising all of the modelling skills suggested above and gave us instant feedback throughout the steps of the modelling process.

I would like to share with you some of my observations by juxtaposing a modelling framework with the Logo polygon construction activity. Presentation of this particular framework for the process of modelling is not to suggest that a lesson plan can be derived from it. In fact, those who teach mathematical modelling caution us that such frameworks not

become empty, ritualistic algorithms into which students attempt to fit problems and their solutions. With this caution in mind, I present a schema of the modelling process in Figure 1. (Mason, 1984) Modifications are based upon information from Cross et al. (1985) and C.E. Lawrence (personal communication, January 1987).

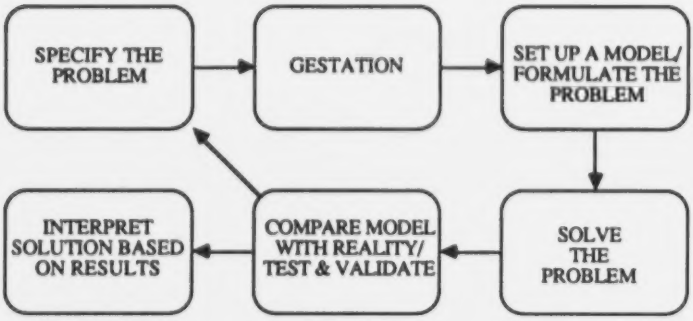

SPECIFY THE PROBLEM
GESTATION
SET UP A MODEL/ FORMULATE THE PROBLEM
SOLVE THE PROBLEM
COMPARE MODEL WITH REALITY/ TEST & VALIDATE
INTERPRET SOLUTION BASED ON RESULTS

Figure 1. Framework for modelling process.

In order to draw a comparison between the experience of the students in the Logo environment and the framework of modelling skills, I will identify the stage in the modelling framework, the Logo setting, and provide additional comment in the form of teacher notes.

Stage I State the problem or question

Logo setting: Problem: Create a squrare using Logo's Turtle Graphics.

Teacher Note: On the first iteration (constructing a square), the problem is posed by the teacher. In subsequent iterations, students begin to formulate their own problem statements.

Stage II Gestation

In this stage, one ennumerates potentially relevant variables, relationships and features of the problem. From this long list, the modeller must select the pertinent features and identify the minimal set of these essential features. Identification of this set of essential features characterizes this as the "aha!" stage.

Logo Setting: Students offer what they know about a square (4 equal sides, 4 corners). Students identify which Logo commands might be used to draw a square and develop a plan for proceeding.

Teacher's Note: Students eagerly begin the model building stage, usually with only sketchy information and vague intuitions about the appropriate variables and their relationships. After a few cycles through the modelling stages, the students realize that more thought is needed, and they begin the process of identification of the essential features. The ultimate use of 90 for the angle is the "aha!" for these 8 and 9 year olds.

Stage III Setting up a model

Logo Setting: Initially, students spend very little time in Stages II and III. They develop a series of Logo commands, for example, fd _ rt _ (using estimated values for each command), which they try in succession, usually 4 times. This is their model.

Stage IV Solving the problem

Logo Setting: The computer gives back the solution by executing the Logo commands.

Stage V Compare model with reality

Logo Setting: Students look at the computer screen and quickly compare the graphics resulting from their Logo commands for their initial model with what they have as a visual/mental model of a square. This is a very exciting and critically important stage. Numerous comments are made. "Wow, look at this. This isn't square!" "Ought-oh, it's not closing!" "Whoops, mine's crooked."

Teacher's Note: Obviously, students are shifting back and forth between stages here. As they compare their model to reality, they recognize that they must THINK about what they tried, what they saw happen, and how this relates to the model of the square they wish to make (back to GESTATION!) During this validation stage, special emphasis is placed on what we have proved and/or disproved, what we have learned by our hypothesis testing, and how closely our model solves the problem or represents the reality. Depending upon whether or not we have revealed the essential features of a square: a) 4 equal sides, b) 4 equal angles, and eliminated the other confusing elements from our model, (i.e. made a square), we return to further testing, or we go on to Stage VI.

Stage VI Interpretation of the solution or results

Logo Setting: Students are now happily reflecting upon their squares.

Teacher's Note: The teacher's role is to bring students to a higher level of understanding regarding the modelling process and its outcomes. Through questioning and discussion, students are able to identify the essence of their model by (1) stating the commands responsible for each part of the square; (2) determining the correct formula and size of angle; (3) recognizing the angle as the critical attribute (essence) of "squareness" in the model. Students are also able to describe those attributes which will not affect the model of the square: size of side, color of pen, and even the number of repeats, (if REPEAT was used).

Stage VI is also very exciting to student and teacher, because it is at this stage that students critically assess their work. They demonstrate that they recognize the essential features when they state with assurity that making the value of the forward command 1000, or 10000, or 1000000, will not alter the "squareness" of their model even though it is no longer visible on the screen. Students further exhibit their understanding of the model for a square when they are able to apply their knowledge of this model to the creation of multiple squares, a set of concentric squares, or a tilted square.

From this point, students expand their understanding of polygons and related geometric principles by experimentation with other models. New problems arise as the process continues and students uncover problems to investigate and model themselves. In subsequent uses of the modelling process, they develop the important modelling skill of moving from particular to general, i.e. from square to pentagon to the general class of all polygons. With

each successive iteration or completed model, students require less direction thus becoming more independent modellers.

In time, these students can be challenged to apply their emerging skill in modelling to real world problems or situations outside the context of Logo. Acquisition of a modeller's perspective is a gradual process, although it cannot be taught in a traditional sense, it can be learned. The teacher, aided by the heuristic environment of Logo, plays the vital role of facilitator and partner in the modelling process. The ultimate pay off is not in the actual models that these children develop and understand, but in the application of their enhanced modelling ability to problems. Equipped with these skills, they will no longer approach life's garden plot problems with only a set of memorized formulas.

Burns, R.W. & Brooks, G.D. (1970). The Need for Curriculum Reform, Educational Technology, 2(1), 8-12.

Cross, M. & Moscardini, A.O. (1985). Learning the Art of Mathematical Modeling. New York: John Wiley & Sons.

Hechinger, F.M. (1986, June 10). Learning Math By Thinking. The New York Times, pp. c1, c7.

Judson, H.F. (1980). The Search for Solutions. New York: Holt, Rinehart & Winston.

Mason, J.H. (1984). Modelling: What do we really want students to learn? In Berry, J.S., Burges, D.N. Huntley, I.D., James, D.J.G., & Moscardini, A.O. (Eds.). (1984). Teaching and Applying Mathematical Modelling (pp. 215-234). New York: John Wiley & Sons.

National Commission on Excellence in Education. (1983). A Nation at Risk: Imperative for Educational Reform. Washington, D.C.: U.S. Government Printing Office.

Rubenstein, M.F. (1975). Patterns of Problem Solving. New Jersey. Prentice-Hall.

Travers, K.J. & McKnight, C.C. (1985). Mathematics Achievement in U.S. Schools: Preliminary Findings from the Second IEA Mathematics Study. Phi Delta Kappan, 66 (6), 407-413.

Educational Computing in the Soviet Union

by
C. Dianne Martin and Rachelle S. Heller

Electrical Engineering and Computer Science Department
The George Washington University
Washington, D. C. 20052

Abstract:

To obtain a full picture of the educational computing environment in the Soviet Union, it is necessary to examine the bureaucratic structure of the educational institutions, the availability of computing equipment, and the potential impact that computers might have on a closed society. The current thrust of educational computing under the new five-year plan is upon the development of general computer literacy and programming skills. A national initiative has been undertaken to provide computer equipment and to develop new computer curriculum materials for the schools at all levels. The Shkolnitza (Little School Girl) computing system is an example of the Soviet effort to provide an enhanced learning environment for students of all ages. The sense of urgency felt by the Soviets to prepare their students for the technological age will provide many opportunities for collaborative research efforts with the West.

Introduction:

Computer education in the Soviet Union has been developed through a top-down approach, starting at the universities with the post-World War II emphasis upon cybernetics. The most difficult problem that the Soviets encounter with any new initiative is the murky water of their own bureaucracies, and the educational computing intiative under the new five-year plan is no exception. Although the Soviets have a national educational system with a nationally established curriculum at all levels, we saw evidence of several different ministries at work, sometimes at odds with each other, trying to shape the new initiative. All educational matters are officially under the direction of

● 1987, Martin & Heller, The George Washington University

the Ministry of Education. The training and certification of teachers, the setting of the national curriculum, and the the overall organization of schools fall under the pervue of the Ministry of Education. However, the Soviet Academy of Sciences also has a powerful impact on the direction of science, mathematics, and computer education with prominent academicians exerting their personal influence to affect changes. This has been especially true of computer education over the past two decades.

Major Thrusts in Educational Computing:

As a result of these conflicting influences there are presently several major thrusts in educational computing in the Soviet Union. Scientists such as Prof. Alfred Aliamazon, Director of the Institute of Problems in Cybernteics at Peraslav, are concerned about the development of good educational software that will integrate computing into all of the subjects for students of all ages. He has established informal computer clubs at his institute to which children walk from miles around on weekends for the chance to play computer games and write their own programs. In his recent visit to the United States he was interested in watching students use computers in science labs, robotics labs, math classes and social science classes. He objects to teaching programming to all students and is concerned about the social and cognitive issues of using computers with children.

A different thrust is being directed in model schools in Siberia by Academician Andrei Ershov, the eminent computer scientist who has been called the "father of Soviet computer education." Ershov helped to shape the computer science curriculum in the universities and wrote the two-year curriculum now being used to teach computer science during the last two years of high school. His curriculum emphasizes algorithm development and programming skills.

Ershov's work with the Shkolnitza (Little School Girl) computing system parallels Papert's work with LOGO in the United States. Shkolnitza is comprised of four parts to be used by students, and they are 1) Robic, which is a LOGO-like language controlling the actions of robots on the screen and designed for children ages 8-12, 2) Rapira,

which is an algorithmic language for senior students, 3) Shpaga, which is an interlanguage subsystem for graphics, and 4) machine-language tools for experienced students. Most of the Shkolnitza system was written under the direction of Ershov by university students. It is Ershov's goal that students who learn the various levels of the Shkolnitza system will then become production progammers who can write good educational and other useful software.

Yet another concern is being addressed by the Pedagogical Institute of the Ministry of Education and that is the training of vast numbers of teachers. In the Moscow region alone, 17,000 teachers receive obligatory inservice training each year. Finding the appropriate people to train these teachers about computers is considered a major problem in the Soviet Union at this time.

Problems to be Overcome:

In addition to the problems of conflicting educational goals, lack of software and lack of trained teachers, the most serious problem acknowledged by the Soviets at this time is the lack of computer hardware. There are 60,000 schools in the Soviet Union that serve a total of 40 million students across 10 grades. Currently, the number of microcomputers in Soviet schools is estimated to be 1,000. The Ministry of Radio and Electronics and Soviet industry has been charged with designing and producing one million microcomputers, of which 500,000 are slated for the schools, by the year 1990. By the year 1995, it is hoped to have one million microcomputers in the schools.

A more subtle problem that the Soviets face, however, is an ideological one. "The computer is a technology particularly unsuited to the Soviet system...Computers undermine secrecy and control, require accurate information, and make that information accessible and transferable. Every computer is a printing press. (Campbell, et al, 1985, p. III-31)". There are also practical problems to be overcome. Few Soviet students learn to type, and the telephone system will not support electronic communication.

Conclusions:

The Soviets have optimistic plans for the next 15 years, leading to major hardware, curriculum, and software developments that will extend computer education to all grade levels. Accomplishing this goal will take major efforts in many areas, most notably in the development and manufacture of hardware and in teacher education. To coordinate all of this activity a special Shkole (School) Committee has been established that cuts across all of the bureaucracies and is under the direct leadership of the new President of the Academy of Sciences, Academician Yevgeny Velikhov.

A positive sign of the Soviet determination to meet these formidable goals is the new receptivity to collaborate with American educators and computer scientists. The recognition that over the past 10 years the United States has dealt with some of the same problems now facing the Soviets in computers in education has spurred the signing of several major research agreements between the U. S. and Soviet Union. These agreements emphasize mutual participation and cooperation between researchers, educators and students of both countries and are seen as a major development in the effort for world peace and understanding.

References:

Beam, Alex. "Russia Gropes for a Way to Enter the High-Tech Age." *Business Week*. New York, NY: November, 11, 1985, pp. 98 - 102.

Brown, Marlene. "Soviet Computing." *Government Computer News*. Washington, D. C.: October, 25, 1985, pp. 56 - 61.

Campbell, R. W., et al, *Soviet Science and Technology (S&T) Education*. McLean, VA: Science Applications International Corporation Technical Assessment Report, October 15, 1985.

Duggan, Dennis. "Computers Warming the Chill." *Newsday*. New York, NY: November 20, 1986, p. 17, 38.

"For Development of a Program for Scientific Collaboration with the USSR Academy of Sciences on the Problem of the Use of Computers and New Means of Communication in School-Age Education (The 'Hamburg-Velikhov Project')." Proposal for research sponsored by the USSR Academy of Sciences, the Carnegie Corporation, and the International Research and Exchanges Board (IREX) signed at the Bank Street School, New York, NY, Oct. 1986.

Ershov, A. P. "Programming, the Second Literacy." Keynote Address, Third World Conference on Computer Education(WCCE). Amsterdam, Holland: North-Holland Publishing Co., July, 1981.

_____ et al. "Computers in School: Formulating a National Program." *Alberta Printout*, Vol.4. No. 2. Alberta, Canada: Alberta Society for Computers in Education, May, 1983.

Hechinger, Fred M. "About Education: Soviet Schools Begin a Computer Curriculum." *The New York Times*. New York, NY: Sept. 5, 1985.

_____ "About Education: U. S. and Soviet to Share Insights on Computers." *The New York Times*. New York, NY: Dec., 1985.

Heller, R. S. and C. D. Martin. "Report on Trip to the U.S.S.R." Project report co-funded by the Soviet Academy of Sciences, IREX, and George Washington University. Washington, D. C. : The George Washington University, July, 1986.

McHenry, William K. and Seymour E. Goodman. "MIS in Soviet Industrial Enterprises: The Limits of Reform from Above." *Communications of the ACM*, Vol. 29, no. 11, November, 1986.

Pollack, Maxine and Ross A. Stapleton. "Why Ivan Can't Compute." *High Technology*. New York, NY: February, 1986, pp. 42-69.

Ulanov, Nicholas A. "Soviet Fear of the Knowledge Revolution." *The Wall Street Journal*. New York, NY: May 13, 1986, p. 30.

STUDENT DESIGNED TUTORIALS

SEAN MCPEAK (13)
KEVIN MCPEAK (9)
JON CAMPBELL (13)

WHISPERING OAKS SCHOOL
R.R.#1
MACKINAW,IL 61755
2/3/87

STUDENT DESIGNED TUTORIALS

BACKGROUND:

WE ARE STUDENTS AT A SMALL SCHOOL IN CENTRAL ILLINOIS. IN OUR SCHOOL SOME OF THE YOUNGER STUDENTS NEEDED SOMEWAY TO IMPROVE THEIR UNDERSTANDING OF THE PLANETS,CONSTELLATIONS, AND TO LEARN BASIC FRENCH WORDS AND PHRASES. SINCE WE HAD BEEN PROGRAMMING WITH LOGO,AND THE YOUNGER STUDENTS HAD BEEN USING LOGO GRAPHICS, IT WAS OBVIOUS THAT BY USING THE COMPUTER AS A STUDENT AID, IT WOULD KEEP THE YOUNGER STUDENTS INTEREST AS WELL AS BE AN AID TO THE TEACHER. WE KNEW THAT THE KIDS LOVED TO SPEND TIME AT THE KEYBOARD. WE ASKED THE TEACHER IF WE COULD WRITE SOME TUTORIALS AND SHE SAID YES. SINCE THE KIDS HAVE USED THE TUTORIALS, WE HAVE FOUND THE FOLLOWING BENEFITS FOR THE KIDS:

1. MORE INTERESTING TO WORK WITH COMPUTER
2. STUDENT CAN ACTUALLY SEE THE PLANETS, CONSTELLATIONS AND FRENCH WORDS WITHOUT A TEACHER NEAR.
3. CHANGE OF PACE FOR KIDS
4. STUDENTS CAN START AT THEIR OWN LEVEL
5. STUDENT LEARNS AT OWN PACE

TUTORIALS/CONTENT:

GALAXY GRAPHICS

THIS TUTORIAL SHOWS SELECTED CONSTELLATIONS AND ALL 9 PLANETS AS WELL AS OTHER SPACE OBJECTS.THIS PROGRAM INCLUDES GRAPHICS AND SOUND EFFECTS TO HELP TELL THE SPACE STORY. ALSO FACTS ABOUT EACH PLANET AND CONSTELLATION ARE PRESENTED.THE OBJECTIVES ON THE PROGRAM ARE TO ALLOW THE STUDENTS TO HAVE A VISUAL IMAGE OF SPACE OBJECTS, LEARN FACTS ABOUT THE PLANETS, AND TAKE AN OPTIONAL TEST ON PLANET FACTS AND/OR SELECTED CONSTELLATIONS. THE TUTORIAL CONTAINS MUSIC AND GRAPHICS .

FRENCH BREAD

THIS TUTORIAL TEACHES THE FRENCH LANGUAGE BASICS. THE YOUNG STUDENT LEARNS SIMPLE NOUNS, PRONOUNS , VERBS, CONJUGATIONS AND NEGATIVE OUTPUTS. THE STUDENT SELECTS THE LEVEL FOR HIS SPECIFIC CHALLENGE. THIS TUTORIAL EXPANDS THE STUDENTS KNOWLEDGE OF THE FRENCH LANGUAGE BASICS, ESPECIALLY NOUNS, PRONOUNS, VERBS, CONJUGATIONS, AND NEGATIVE OUTPUTS. THE LEARNING IS AIDED BY MUSIC, AND GRAPHICS WHICH AID THE LEARNING.

REFRENCES USED

GENERAL LOGO:
THE TERRAPIN LOGO LANGUAGE FOR THE APPLE II, VIRGINIA CARTER GRAHAM,PETER VON MERTENS, TERRAPIN INC. CAMBRIDGE,MA,1982

LEARNING WITH LOGO, DANIEL WATT, MCGRAW HILL BOOK COMPANY,1983

LOGO PROJECT BOOK, ALISON BIRCH,TERRAPIN INC., CAMBRIDGE ,MA 1986

GALAXY GRAPHICS:
OUR UNIVERSE, ROY A GALLANT, NATIONAL GEOGRAPHIC SOCIETY,SPECIAL PUBLICATIONS DIV. 1980

ON THE BRINK OF TOMORROW-FRONTIERS OF SCIENCE, NATIONAL GEOGRAPHIC SOCIETY,WASH.D.C.,1982

FRENCH BREAD:
VIVE LE FRANCIAS, G.ROBERT MCCONNELL, ROSEMARIE GIROUX COLLINS, RICK PORTER, ADDISON WESLEY, 1978.

HOW WELL SUITED WAS LOGO FOR TUTORIALS?

WE FOUND ONLY ONE AREA FRUSTRATING AND THAT WAS PICS. WE FOUND A NEED TO EDIT PICS EXTENSIVELY ESPECIALLY IN GALAXY GRAPHICS. WE FOUND OURSELVES LIMITED IN THE EDITING AND CHANGE PROCESS. IN ORDER TO EDIT WE HAD TO COMPLETELY RE-DO THE PIC STORED ON DISK. THERE HAS TO BE A BETTER WAY.

FUTURE PLANS FOR TUTORIALS:

WE PLAN TO EXPAND THESE PROGRAMS WITH MORE MUSIC,GRAPHICS, AND QUIZZES. ALSO WE PLAN TO USE WORD LISTS AND ADD MORE LEVELS OF DIFFICULTY TO EACH TUTORIAL. WE HAVE ALSO BEEN WORKING ON A SPELLING TUTORIAL FOR THE YOUNGER STUDENTS.

WE THANK YOU FOR LETTING US PARTICIPATE IN THIS CONFERENCE.SINCERELY,

SEAN & KEVIN MCPEAK
BOX 229 R.R. 1
MACKINAW, IL 61755

JON CAMPBELL
BOX 230 R.R. 1
MACKINAW, IL 61755

"Tools for Making Meaning out of Mathematics" -A Rationale

Harold R. Melnick
Bank Street Graduate School of Education

I've been invited to speak with you today not as a LOGO expert , though I am delighted with the language, but rather because I am told that I help teachers design curriculum which is "LOGO-like". I find that an interesting thought , though no one has yet defined for me what exactly that means. I do know what it is that I try to do with teachers, what other math educators in the country like Marilyn Burns, Mary Laycock, Jim Bruni, Patricia Davidson , and a host of elementary teachers are doing to affect change. I thought then for this talk I'd examine one way we are tackling the task of matching student's learning capacity with what mathematics really is ;through the use of tools that enable children to attach meaning. Perhaps by the end of this talk you can help me better understand how this approach I speak of is or is not worthy of being called"LOGO-Like".

I'd like to start first by reminiscing some. I remember when I attended an introductory course in LOGO some 5 or 6 years ago I sat down to the computer with a partner who was a 4th grade teacher in New York City. We were given a few introductory commands and were told to play around with the turtle and see what we could make the turtle do. With great excitement, I suggested to my partner that together we try to teach the turtle to make hexagons and then figure out how to make a tesselation of hexagons , repeating in in all directions, covering the whole screen. I drew a quick sketch of what I was saying. What fun I thought..... until I noticed a look of despair on my partner's face.

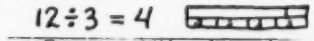
"What in heavens name is a tesselation?", he questioned, " and besides is this hexagon thing like a stop sign shape? Can't we start with shapes I know like triangles , squares and rectangles? Besides I hated geometry in school." I was a little stunned but pleased with his honesty. We moved to making the turtle play with squares for a bit and then I had an idea. Partly due to the inherent teacher in me and partly due to my fixation on getting the hexagons to tesselate, I ran to my office , leaving my partner happily engaged in squares and returned with a bag of brightly colored wooden Pattern Blocks. The hexagons are one sized and are yellow, the triangles are green, rhombus' are blue and are tan , squares are orange and the trapezoids are colored red.

Upon my return my partner immediately started sorting them and playing with them by making patterns and designs. Suddenly he suggested that we reproduce a pattern we made right on the screen. He noticed that the yellow blocks were not quite stop sign shapes; that they had six sides. He asked if they were the hexagons. I smiled. He talked with me about how nicely the shapes in this set all fit together . There was a satisfaction about his comments that was a nice contrast to the earlier look of despair. That brought us to looking at the equivalent angles and side lengths and the notion about degrees in an angle. I showed him what a tesselation was and he equated this to the work of M.C. Escher and Islamic Art. He was delighted with the blocks and he designed a new tesselation of hexagons and triangles which was much more interesting than my original plan. The real material seemed to provide him the concrete experience which grabbed his attention. After much playing and discovering about angles and side lengths , he and I programmed our turtle to tesselate and we were both delighted with our work.

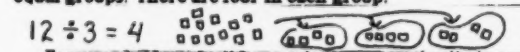
copyright © 1987 by Hal Melnick

I cite this example to point out that while LOGO is surely a simple and compelling medium to ask children (and their teachers) to think about their mathematical reasoning, I think there is a companion requisite; prior experiences with many 3 dimensional mathematical relationships. These experiences serve as a basis from which the learner takes ideas and represents them and expands upon them in the two dimensional medium of paper or on the monitor's screen. There is also something to be said for the sensorial memories as strengths which many of us have that are oftentimes underutilized during the schooling years. Three dimensional materials which are designed to reflect geometric or numerical relationships are tools for making meaning out of mathematics. Even though every child does not require them to see and understand the underlying relationships and concepts, ...many children do. And too few children in this country are privileged enough to have them to do math with .

Too many children are still learning division as the "GUZZINTO" process, where twelve divided by 3 is thought of as "3 GUZZINTO 12". Too few children have the opportunity to build a representation of twelve with a ten rod and then a two rod, and then through a process of trial and error , finding out how many threes can be measured (or fit) underneath the twelve. " How many threes really do fit under the twelve? " is the question the child asks . "Four of 'em." is the answer the child constructs .



Too many children learn about the other way to tackle a division problem only when they are faced with silly word problems that require that they recall a process they may never have thought about before. This other way is called the Partitive Approach or the Sharing Way. Twelve divided by three is here viewed as the collecting of twelve things and the sharing of these twelve things into three equal groups. There are four in each group.



Because children don't have extensive opportunity to contruct the relationship of division with real things , this operation oftentimes becomes a child's downfall. This results in limited ability to apply the facts. When we hear children say "Is this a 'divided by' problem?" they most probably haven't had adequate opportunities to physically share or measure off divisors!

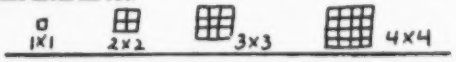
Manipulative materials like Cuisenaire Rods, Pattern Blocks, Powers of Ten blocks, Unifix or Multilinks and others are not to be used as visual aids to just show children what they are writing in abstract form on the printed page. Manipulatives, to be lastingly useful are given to students as tools to solve their mathematical problems. They may be used to see the relationships, to feel the relationships, to construct the ideas (abstractions) describing those relationships.

It is this work that I am doing - working with teachers, administrators, parents and school systems to understand the value of designing 3 dimensional learning opportunities for all children, so they may grow to see mathematics for what it is, "a study of relationships between objects or quantities that have been organized so facts may be proved or derived from others using logic."

The long term affects of never understanding the meaning of symbols taught in school may be best illustrated with this anecdote:

When teaching some early LOGO programing to three teachers during a math lab class in the course I teach, they reached an impasse when trying to draw a right triangle to fit a design they were making with the turtle. I asked if they recalled anything about the relationship between the two legs of the right triangle and the hypotenuse. Two of the teachers hadn't a clue and the third said that she'd remembered a formula $A^2 + B^2 = C^2$. "But what does that say about relationships?" I decided to reroute to find out what these adults knew about squaring numbers (or what it means to raise a number to its second power)

Using unifix cubes ,I placed one square on the table and called it a one by one square floor (1X1). I asked them each to try to build a bigger square floor, any number of units long. We laid them all out.



This was the first time each teacher was ever shown the geometric explanation for why 3 squared is 3X3. They were each flabbergasted by the physical relationship between the number representation and the concrete physical representation of something they had always believed to be purely an abstract idea.

One woman suddenly grabbed at the cubes and started to build something 3 dimensionally. "Look", she said "if these things are called square numbers then cubic numbers.... I'll bet are cubes!!! " This construction time went on for a bit as the three danced through this wonderful "Ahah" experience together. The lightbulb had been lit within the minds of these three adults ; three bright, intelligent , competent adults who had all their lives in school been starved of real mathematical experiences with 3 dimensional meaning making tools.

3 cubed is really an abstract recording of a cube whose length is 3, height is 3 and depth is 3.....3 used as a factor 3 times!

I felt that we could now go back to looking at the relationship between the sides and the hypotenuse of a right triangle . After asking each of the teachers to build a square the length of each side of the triangle, they jumped at making conjectures about the relationship between the numbers, concluding that the square of side A added to the square of side B is equivalent in area to the square of side C.

$A^2 + B^2 = C^2$

$9 + 16 = 25$

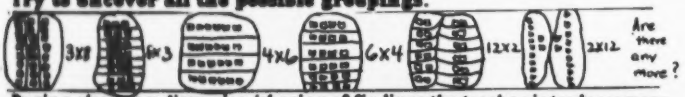
I handed them each a Geoboard with a rubber band right triangle on each and asked them if this relationship is true for all right triangles . They fiddled with this for a while. Utilizing varied medium to validate the same point is important especially since some manipulatives do not make sense to some people while others show the ideas much more clearly. $A^2 + B^2 = C^2$ became real to these teachers for the very first time. These three had begun to understand a geometric relationship that they then could apply constructively to their work with LOGO, (which they proceeded to do).

Research indicates that young children learn well when new ideas are presented to them in context. Children are good problem solvers when they are given time to think and can discuss their ideas with friends. They are also quite capable of coming up with inventive solutions when they have not previously been overburdened with rules and procedures to follow for a given problem. Children understand when they are provided the concrete tools to help them reason.

Children taught in a manipulative based 2nd grade experiential program, instead of being given multiplication tables to learn immediately, might be introduced to multiplication by first being challenged with a focussed thinking question asking them to use counting cubes to show what they are thinking.

How many different ways can you arrange twenty four cubes into equal groupings?

Use blocks to show your findings . Then record your findings . Try to uncover all the possible groupings.



During classroom discussion/sharing of findings the teacher introduces ways to formally record the groupings using the "X" to stand for the words "of the " as in "3 of the eights". We have found that this terminology is far better to insure understanding than "is multiplied by" or "times by" when introducing the sign because those last two terms seem less related to the context of the grouping experience.Some students immediately notice that 3X8 is another way of stating 8X3 while others do not see that for some time. This gives the teacher important diagnostic information about about the children's readiness to abstract generalizeable principles from concrete evidence. That ,in and of itself, reflects a stage of growth which many 2nd graders have not yet achieved.

The face of mathematics education in this country is changing- it must change. Each of you attending the Northeast Regional Logo Conference is well aware that our children are growing within an "Age of Information "as referred to by Howard Gardiner in Frames of Mind What is most needed today is a population of flexible thinkers equipped to use the wealth of information available to "invent" and to "creatively solve problems". However, Vincent Rogers in "Exceedingly Effective Schools, (Educational Leadership, April , 1983) contends that our schools claim to educate broadly but , in practice , emphasize a narrow "three R's curriculum taught to large groups of children in a relatively standardized way using textbooks workbooks , ditto sheets , and teacher dominated talk as principal techniques."

Many children and adults have been taught to do math and have never been taught to think math. Students who have the greatest difficulty with mathematics are usually those who attended classes where they were taught rules for doing certain kinds of problems. They believe there is only one way to do a long division problem , that you may only do a subtraction with regrouping problem (572 - 194) by working from the ones column through to the tens and last the hundreds, that 1/2 X 1/3 merely means" multiply the tops and then the bottoms and bring them across", and that 1/2 divided by 1/6 requires that. you invert and multiply.....but why do you invert the divisor fraction and switch the operation to its inverse, multiplication?.....Oh yes we are also taught in math that...."yours is not to reason why, just invert and multiply!" The problem is that students memorize these *"only one ways"*. If their memory fails them and they find that on a test they cannot remember one step in the memorized algorithm (the procedure) they find themselves unable to solve the problem.

The result is that students are not flexible in their reasoning and deal poorly with non -routine problems and have difficulty coming up with a alternate solutions. Many students are leaving school with limited abilities to reason numerically, spatially and logically, not because they are incapable , rather because of the mathematics programs which have deprived them of meaningful thinking

With Problem Solving as the focus of math education, we are seeing shifts occur. More and more school systems are adopting the broadened definition of Basic Skills that was developed by the National Council of Supervisors of Mathematics. Redesigned State Curricula are beginning to reflect the Ten Areas defined which are:1) **Problem Solving** 2)**Applying Mathematics to everyday situations**. 3)**Alertness to Reasonableness of Results** 4) **Estimation and Approximation** 5) **Appropriate Computation Skills** 6) **Geometry** 7) **Measurement** 8.)**Reading, Interpreting and Constructing Tables, Charts and Graphs** 9) **Using Mathematics to Predict** 10) **Computer Literacy.**

Texas has mandated the use of Manipulative tools in all classes K-12 to introduce new concepts in math. North Carolina has followed with a similar set of requirements and California has published a dramatic new document calling for sweeping changes in the predominantly paper and pencil , rule oriented treatment given mathematics by the textbook and workbook publishers.

As you probably know , as of the writing of this paper , the California State Board of Education has rejected all 14 traditional math textbook series submitted for state adoption. The textbook publishers have been given one year to revise and resubmit texts. The revisions called for are in three areas; 1)Understanding, 2)Problem Solving and 3) Number Sense. Students are expected to be asked to use manipulative materials from which they can make their own explanations and generalizations by talking with other students. This emphasis on teaching for understanding is long overdue but surely welcome. The new texts are expected to give students opportunity to apply their math knowledge to new challenging problem solving situations. And with regard to number sense, instead of focusing on computation facility heavily emphasizing time to practice writing algorithms , the programs are expected to aid students in developing a strong understanding of numbers. Children are to be asked to choose their most efficient method for computing- on paper, mentally, by estimating, or with a calculator; the focus being on the student becoming responsible for finding his/her best methods. In addition students are expected to consistently explain why their answers are reasonable.

Looking back through the evolution of mathematical thinking, Pythagorus and his colleagues would conjecture, test , experiment, retest and challenge their own thinking. Young children and their teachers could benefit immeasurably from simulating that experience. As Mary Laycock said at the last NCTM National Convention "you know, mathemeticians spend about 90% of their time thinking and about 10 % of their time writing . Kids in schools in this country spend about 90% of their time writing and 10% of their time thinking during their math lessons in school". This is quite a statement.

If mathematics continues to be taught as a bunch of rules to be learned , then children will never fully see how it really is a tool to help us think , solve problems and forge ahead with new ideas. If we expect children today to be creative users of mathematics they must participate in discovering what the world of relationships is about. Through an experiential mathematics program filled with skillfully used manipulatives - children can develop as strong , adaptable mathematical thinkers.

Apropos Logo and Telecommunications

Robs John Muir
Claremont Unified School District
Claremont Graduate School
Claremont, California

From EMSIE (Educator's Information Exchange) via BIX (Byte Information Exchange) via 1200 baud modem.

The message is the medium:

```
telecom/jan.discuss #12, from gwigley, 2537 chars, Tue Oct 21 17:54:12 1986
-------------------------
TITLE: Comments from Robs Muir

The following comments are excerpted from a special Logo conference on BIX,
Byte magazine's computer conferencing system.

Robs Muir is a physics and computer science teacher in Claremont, CA and an
instructor at Claremont Graduate School.  He is a regular columnist for the
Logo Exchange.

logo/experts.forum #8, from rmuir, 5034 chars, Fri Jul 25 23:49:35 1986
There is/are comment(s) on this message.
-------------------------
TITLE: Regarding telecommunications...

The Logo Conference Tree project, funded in part by a grant from the
Apple Educational Foundation has just concluded the second year of its
grant.  Since Mark Guzdial mentioned the interest in conferencing at
Logo '86 in an earlier remark, I would like to make several broad
observations about teleconferencing, educational systems, and the "Logo"
philosophy based on my experience as project coordinator of the
Conference Tree.

The Logo Conference Tree project established a computer based tele-
communications system in Claremont, Califonia which linked seven
elementary schools, a junior high, a continuation high school, and an
orthopedically handicapped school in a tree-structured conferencing
environment.  Using conventional hardware (Apples, floppy drives, and
300 baud modems) and CommuniTree software (a Forth package written by
the CommuniTree Group of San Fransisco), the project focused on
building a transportable conferencing model targeted for educational
settings.

Unlike typical BBS or information services (CompuServe or BIX) or even
BITnet or ARPAnet, CommuniTree is an attempt at an _anonymous_ conferencing
host.  No menu-driven command sets here; instead CommuniTree requires only
two commands to have nearly complete access.  Namely, READ <messagename>
and ADDTO <messagename>.  No passwords or userIDs are required and the
system enforces (encourages?) hierarchical messaging.  Any user can add
a submessage at any node and searches can be initiated on any branch.

One of the more interesting aspects of the project was that the Logo con-
```

ferencing systems (we ultimately had three different hosts in operation) could be accessed from home computers. Coupled with the anonymous nature of Tree conferencing, these systems were a link between schools and the larger community. Indeed, several conference strands involved a wide variety of age levels, from third-graders to senior citizens. A samba school for the modem set. This heralds what schools might be, except for one hitch...

(more in next message)

---

telecom/jan.discuss #13, from gwigley, 2935 chars, Tue Oct 21 17:55:31 1986
There is/are comment(s) on this message.

TITLE: More from Robs Muir

The school terminals were rarely used during class time!

The systems maintained an daily average of 350 messages, but most of the active participation was from outside to the schools. This, despite the installation of terminals in every school and extensive inservices given to each school faculty.

I have several pet theories for this. Let me list some of them.

1) Telecommunications represents a quantum leap beyond where teachers are today. With rabid technophobia and rampant technotimidity the norm in the adult population, teachers are unwilling (unequipped?) to appropriate new tools in the schooling process and are, hence, disinclined to present these ideas to their students.

2) The teaching profession is already hopelessly overburdened with too many responsibilities. Teachers cannot be expected to embrace a new domain of knowledge, i.e, computers, on top of the 37 zillion other things they must do each day. Have you taught second-grade lately?

3) Telecommunications requires too much individual time. Only those who have direct, extensive access to communication channels can effectively use them. Home use--where there is more unstructured time--is the best means to tap such resources.

4) Extensive use of computers in schools will not be achieved until a threshold is reached. It may be "one computer per child". Or, in telecommunication's case, one may need "three friends on the network" to make the experience meaningful and engaging.

5) The classroom, with its overly busy schedule and exhaustive curricula, has no room for individual activities such as telecommunications. The herd is more important that is each cow (sheep?). Although classrooms often appear to isolate students--one worksheet per student and no talking!--
in fact, the forces all conspire to insure that everyone does the same thing. Its easier for a beleaguered teacher to manage 32 kids that way.

6) (My favorite.) Schools are, a priori, incompatible with the free exchange of ideas between peers. In fact, one county agency refused to participate in the project because they had no way of monitoring the usage of an anonymous conferencing system! The system is threatened by

too much liberty.

The Logo Conference Tree was intended to provide a support environment for users of the Logo language as well as support a wide range of other curricular areas. To a limited extent, it was successful.

As a precursor of the wider uses and potentials for computer telecommunications in education, the project was either "before its time" or there never will be a time. At least for schools...as we know them.

In some ways, I an inclined to feel this way about programming in schools. There really won't be any room for it until there is a major restructuring of the schooling process. Until that time, some of the most interesting programs will be written by students _after_ they get home.

No Threshold? No Ceiling?

Robs John Muir
Claremont Unified School District
Claremont Graduate School
Claremont, California

During the past Summer, I had the opportunity to offer an introductory course for the Teacher Education program at Claremont Graduate School (CGS) in Claremont, California. Titled "Innovative Uses of Technology in the Classroom", this course was designed to provide an intense 21-hour introduction to the potential application of emergent new technologies for Piagetian, or child-centered education. While Logo has always been a focus of this course during the past four years, based on my personal experience with LogoWriter I made the decision to use LogoWriter as the vehicle for the 1986 course rather than Apple or IBM Logo; the risk was entirely mine. After all, innovative software demands innovative uses!

The student profile of the interns was very representatives of any population of well-educated adults. Some owned their own computers and were capable programmers (BASIC, Pascal, and COBOL), while other students exhibited the technophobia we computer educators have come to expect. The only commonality in the group was their desire to teach, their committment to children, and their lack of experience with LogoWriter.

I suspect that these students have been among the first teacher interns to use LogoWriter in the United States.

My reactions to LogoWriter have been forged in (under?) the fire of exploring a programming environment admittedly designed and targeted for middle-school. If I had any reservations about switching from vanilla Logos to LogoWriter, my concerns were chiefly with the hazards of using a "juvenile" software package with high-caliber, post-graduate students intent on teaching in all variety of classroom settings--from astronomy to zoology, kindergarten to 12th grade. In all, I had a total of 49 students, in two sections, that met twice weekly for four weeks! Intense is a good adjective.

While the result of teaching this course was a decided "mixed bag" (I would advocate spreading the learning over a longer period of time.), the range of interesting student projects were little short of staggering. The LogoWriter environment clearly offered some advantages over traditional Logo systems. The approach highlighted by LCSI's project booklets and activity cards helped spawn a host of student projects with a much wider variety of applications, ranging from interactive demonstrations of Boyle's Law to desktop publishing for students, from animation projects to inquiries into poetry and language.

I am an advocate of the computer as an appropriate learning environment unto itself. Computers need not be a vehicle for delivering curriculum; they can become a curriculum at least as rich as any other...if Education will allow it. It seems to me that computers can offer a user the same benefits as drama, music, or sports might offer another someone else. For students (and other users) interested in using the full power of their machines, LogoWriter--in its current implementations--has some limitations. Chief among these are: limited program development workspace; extremely restricted I/O; no documented access to machine language "hooks"; a more restricted primitive set; a non-standard, undocumented operating system (at least for the Apple version) precluding file I/O and telecommunication transfers.

While providing some important new tools for certain segments of the Logo community, LogoWriter does nothing to debunk the myth that Logo is a "children's" programming language. The materials provided with LogoWriter preclude it from ever being used in a high school computer science department, especially if the department chair is big on AP Pascal!

During the early years of Logo at MIT, many grand ideas for Logo were possible since Logo only ran on large, expensive machines. To shoehorn Logo into small, personal computers, parts of Logo had to be eliminated. This was very evident in TI-Logo and early MIT Logos. LogoWriter has helped redefine and extend a bit of Logo's expressive power for those at the lower thresholds. For those who wish for "no ceiling" for Logo, I'm afraid we'll have to wait for a more comprehensive implementation.

Or perhaps, a new generation of machine.

Secondary Logo Opportunities

Robs John Muir
Claremont Unified School District
Claremont Graduate School
Claremont, California

Logo has had a difficult birth. Its childhood spent in elementary school certainly hasn't been without controversy. But, thankfully, Logo has grown to like grade school. Now that Logo has made it to high school (well, some high schools...), we continue to have doubts about its future. Can Logo survive in a single-subject teaching environment? Will the turtle graduate to college? Or will Logo instead go on to build a career in business?

I intend to examine the role of Logo in the Secondary classroom—the problems as well as the successes of high school Logo, and the possibilities for the future. However, to stretch this paper beyond the intended scope of its title, I would like to propose that there are also Primary and Tertiary opportunities, as well as Secondary opportunities for Logo.

In today's schools, Logo is used principally as a tool for computer programming and computer awareness. While this was clearly one of the important roles that Logo was meant to play, the manner in which many schools have integrated Logo into the curriculum has "short circuited" some of the design goals of the original Logo Project. One of the motivating forces behind the original development of Logo was the perceived need for educational reform, or more accurately, educational revolution. The focus of much of MIT's early research projects was on the learner. Logo was perceived as "child-centered" first; curriculum delivery was of secondary importance.

The main uses of computers in today's secondary schools can be quickly summarized in the following outline. While by no means is this to be considered a definitive list, it has the major components most educators would cite as components of a model curriculum.

1. Wordprocessing
   - Personal use
   - Desktop Publishing
2. Programming courses
   - BASIC
   - AP Pascal
   - Logo
3. Data management
   - Spreadsheets
   - Data Base Management Systems
4. Other
   - Graphics & design
   - Telecommunications

A major emphasis, currently, is on integrating computers into the curriculum. The growing belief is that computers should take a back seat to the existent curriculum; in this view, computers are merely a tool to achieve an end. They are not an end in themselves. (In my own cynical view, this can be seen as a ploy to resist educational evolution and to take the fun out of computing.)

In spite of all the ballyhoo about integrating computers into the curriculum, schools continue to view computer education as an isolated segment of the curriculum. Much of this resistance to integration is due to educational inertia. This inertia has several origins,

namely, well-established teachers continue to offer traditional computer offerings, and limited resources—both in qualified staff and computing resources.

The perceived hierarchy of computer use in high school education in 1987 still follows these three "traditional" levels.

PROGRAMMING — the Primary computer use:

> High schools have tended to use Logo (if at all!) as, primarily, a computer literacy/computer programming tool—a way of introducing students to computer languages and applications. Computer science offerings in high school still tend to "teach a language", rather than teach computer science. Issues of problem organization, data structures, and computational efficiency can be successfully taught using Logo or Pascal.
>
> Computer programming offers more than just the application of a bizarre, arcane syntax to a problem domain. As John Allen states, computing is "mind-training." In years past, learning Latin was used as a form of sharpening mental discipline and clarifying expression. Mathematics serves this role in some fields. Programming (at least using an expressive language such as Logo, LISP, or APL) can offer these benefits with exciting, dynamic, and interactive feedback.

DEMONSTRATION — the Secondary computer use:

> Secondarily, computers are often used as a vehicle for demonstrating or illustrating concepts of an existent curriculum. Here, I'm thinking of the large-screen monitor connected to a computer in a Physics or Mathematics class. I can't think of a more inviting environment to build simple models or simulations of concepts than Logo. Several hours of my physics course are devoted to demonstrating the basics of Logo programming. This becomes the basis for a myriad number of presentations.
>
> The burden here is on the instructor. While the computer is integrated into the curriculum, a knowledgeable teacher is a necessity. Directed teaching is the norm in this use; student-centered education takes a back seat.

PERSONAL LEARNING — the Tertiary computer use:

> Finally, computers are sometimes (rarely) used to give a student a means to build an individual identity, to bolster self-esteem. Much of the literature in the Logo culture focuses on this issue. Logo can be an appropriate vehicle for self-expression—equally as valid a medium as drama, music, or athletics. High schools, with their emphasis on academic preparation, tend to place this aspect of a student's learning experience near the bottom of the list of priorities.

It is my belief that many of us at the secondary level have Logo's priorities backwards. In Logo, we often have need for a REVERSE operation. It often looks like this:

```
TO REVERSE :list
 IF EMPTYP :list [OP :list]
 OP LPUT (FIRST :list) (REVERSE BUTFIRST :list)
END
```

This recursive operation works in this fashion:

? SHOW REVERSE [A B C D]
[D C B A]
? PRINT REVERSE [PRIMARY SECONDARY TERTIARY]
TERTIARY SECONDARY PRIMARY

In designing the environments our children will experience as they go through school, I think it is appropriate for us to consider REVERSING the current order of priorities. The humanity of the learning process must come first, computers need to be seen in functional roles within a curricular discipline demonstrated by competent experts, and finally, the computer as an instrument for sharpening the mind should not be neglected.

Logo can serve as a powerful tool in each of these three areas. As an affective environment for exploration and personally structured learning, Logo epitomizes software with "no threshold and no ceiling." At the secondary level, Logo can provide a positive computer experience for special populations who are often unsuccessful in traditional classroom settings. Logo should be offered as an "elective"—a self-selected course of study without prerequisites. Piaget's view of cognitive development is as equally valid for 17-year olds as it is for the preschooler; students should have the freedom to acquire knowledge on a personal basis. The implication here is that computer programming is for everybody, not just for those who have taken Algebra I.

Many possibilities exist for Logo in today's high schools, yet most of these possibilities remain untapped. We are keenly aware of the Turtle Graphics domain; this is the stereotypical image of what a Logo user does. Among the wide variety of applications that Logo can be put to in a high school, I would like to list several new (and not so new) uses for Logo. Here are a list of the New Possibilities:

- Reference-based writing (a la Hypertext) using Logo as the command language. This takes wordprocessing into new realms, especially in the preparation of research papers as might be found in a History or Government class.

- Language microworlds for students to actually DO science on the syntax and structure of a natural language, such as English.

- French, Spanish, German, or Arabic(!) versions of Logo to reinforce the study of foreign languages.

- Animal behavior simulations in Logo as a tool for understanding biological systems.

- New issues in computer science, such as object-oriented programming paradigms, are now possible using "shells" written in traditional Logos, or object-oriented programming has been highly optimized in new products such as ObjectLogo.

- With advanced Logos, such as ObjectLogo, concepts appropriate to the mathematics department are now not only possible but are opening exciting new possibilities. Among these concepts are: Real and Complex number computation, Matrix operations, and Lambda calculus.

As Logo matures, and as new technologies drive the cost of hardware ever lower, we expect to see more powerful applications for a programming language in education. Education cannot afford to take the "easy" road to computer literacy by simply teaching to the machine or forcing the computer to limit its role to a mere delivery system for "traditional" education. Logo has the power to reshape the face of the schooling process. The challenge rests not with the students—they will absorb the new ideas. The challenge is in the hands of the facilitators. Can we order our priorities for educational change appropriately?

Teaching Methods of Interfacing Logo Procedures

Dan Novak
Ithaca College
Ithaca, New York 14850

The methods and ideas presented in this paper were tested successfully in a college level computer literacy course. A language similar to Logo, which supports operations on procedures (pictures) was used, but most of the problems could also be programmed in regular versions of Logo.

The key idea of this paper is a way to teach students that the starting and ending position of the turtle in the picture (procedure) they programmed effects the way that picture can be used as a building block in more complex pictures.

Uri Liron (The Computing Teacher, 12(5), p. 30) has defined the problem as a lack of clear concept of the interface or the "seam" between procedures. In the following examples and exercises the letter "s" will denote the starting position and the letter "e" the ending position of the turtle relative to the picture being created. In this paper we suggest a series of exercises that can help students understand these concepts by openly and consistently making the interface to be part of the procedure which is being created.

Example: The procedure SQUARE1 was created by:

```
TO SQUARE1
FORWARD 50 RIGHT 90
FORWARD 50 RIGHT 90
FORWARD 50 RIGHT 90
FORWARD 50 RIGHT 90
END
```

The command: REPEAT 4 [SQUARE], when executed, would just redraw the same square 4 times.

On the other hand, one can define the procedure SQUARE2 by:

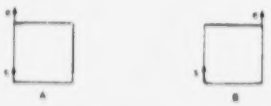
```
TO SQUARE1
FORWARD 50 RIGHT 90
FORWARD 50 RIGHT 90
FORWARD 50 RIGHT 90
FORWARD 50
END
```

This time the command: REPEAT 4 [SQUARE2] will produce a "window like" design:

Even though SQUARE1 and SQUARE2 look the same, the turtle's heading is different when the drawing is done. In SQUARE1 it is the same as it was when it started, but in SQUARE2 it is not the same.

Exercises:

1) The teacher prepares a family of basic designs like the following:



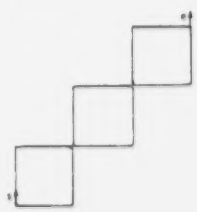
Then, as a first step, the student is asked to draw pictures that correspond to sequences of commands, such as AAA or BAB (this can be done in various forms: worksheets, blackboards, homework exercises or even body movements). The student can later on try to see if his/her answer was correct by checking it on the computer or the teacher can use the computer to demonstrate some of the solutions in class.

Question: A A A

Answer:

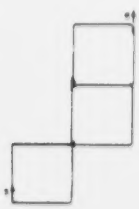
Question: B B B

Answer:



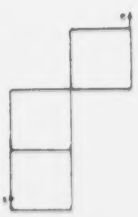
Question: B A B

Answer:



Question: A B B

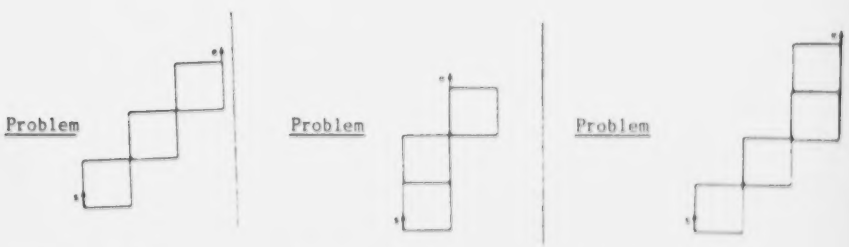
Answer:



Note that even a simple basic design like a square leaves room for many different examples. In particular, if we limit the initial and final positions of the turtle to the corners of the square, allowing the heading to be only whole multiples of 90 degrees we get 16 x 16 = 256 basic designs for the teacher to choose from. Here are some additional examples:

On the other hand, if we do not impose any restrictions on the initial and final states of the turtle, we get a very large number of procedures.

2) The second step is to present the student with pictures and ask him/her to produce the sequence of letters (limited to the procedures that were predefined and in our case to the procedures A and B) that could be used to produce the picture. For example:



Answer: B B B

Answer: A B A

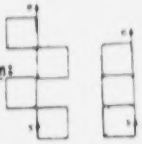
Answer: B B A B

3) This third stage is harder then the previous ones: The student is given a family of pictures and is asked to build a group of basic building blocks that can be naturally used to build these pictures. Many different variations and degrees of difficulty are possible: The teacher can limit the basic building blocks to be of a certain kind, allowing only squares or triangles, but not limiting the starting and ending positions of the turtle relative to the square or the triangle. Note that the answers are not unique and our suggestions represent only one possibility.

Problem:

A possible basic block:

Problem:



Two possible basic blocks:

Problem:

Two possible basic blocks:

A

B

C

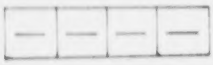
D

E

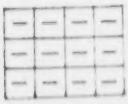
4) A more difficult problem would be to give the student a complex picture and simply ask the student to create it in some natural way, by trying to break the problem into smaller parts.

Examples:

1)



2)



3)

LET THE VALIANT TURTLE ROBOT LEAD THE WAY TO LOGO!

Barbara A. Nye, Ph.D., Executive Director
Tennessee State University
Center for Training & Technical Assistance

in conjunction with

Andre Rossi, Director, Ed. Marketing
Harvard Associates, Inc.

This conference presentation will consist of an actual demonstration of the Valiant Turtle, furnished by Harvard Associates, performing some of the classroom activities recommended by the Tennessee State University Center for Training and Technical Assistance to introduce LOGO in pre-school and elementary classrooms.

Let the Valiant Turtle Robot Lead the Way to LOGO!

The two red eyes light up and a soft whirring can be heard as the wheels spin and slowly propel the green and beige turtle forward. Beneath the see-through green plastic shell lights on the hexagonal computer card go on and off and colorful wires connect to the wheels. The turtle rolls across the floor and stops right in front of an obstacle. Cheers go up from the children clustered around the turtle. They have correctly estimated the distance for the turtle to go without running into the pond they have drawn on the turtle's course.

Now, the teacher asks, how can Val get around the pond? It can't go straight ahead -- it would be in the water. Turtle robots are electronic; they can't swim. So how can it get to the other side of the water? A child demonstrates how he would go around it by turning left and walking around the edge of the pond.

The teacher asks the class to form the instructions which Val will need to do the same thing. How much of a turn does Val need to make? 10 degrees, 30 degrees, 45 degrees, 90 degrees? They decide to try 30 degrees and the instructions are typed in by a student at the computer keyboard. They watch Val execute the turn.

The teacher asks if that's enough, or is it too much? They can adjust the angle if needed. Now, which way does it need to go? Forward? How many steps? Let's try 20. The turtle moves forward, but it's dangerously close to the water. Do we need to adjust its direction? Yes. Which way? Right or left? Left calls out the group. How much of a turn? Not much, maybe 20 degrees. The course is altered by the command on the computer and Val turns slightly to the left.

Now, the turtle can go forward again. How far this time? 50 turtle steps. Then another course correction to the right 20 degrees, and 50 more turtle steps, and soon Val has arrived on the other side of the pond and can continue the trip home. How many steps? Which way does it need to turn? Finally the turtle robot glides into the fancy cardboard house which the class has made for it. Home at last. The class cheers. They have helped Val through the obstacles of the Turtle Town they have designed and back to the house.

Another day, the class is designing a box so they can wrap up a present for Val. What shape is a box? How can we draw a square? Make sure Val's pen is down so that we can

see what we're drawing. Forward 30. Now a right turn. How many degrees? It has to be 90 to be a square! Now, how long should the next side be? It has to be the same size as the first side to be a square! So Forward 30 again. Right 90, Forward 30, Right 90, Forward 30. And the class has made a big box for Val's present. Now let's put a bow on top of the box. What shape can we use to make a bow? Two triangles pointing at each other! Great, how do we make a triangle? 3 sides of equal length with right turns of 120 degrees in between. How long should we make the sides of the triangles? The class decides that each triangle should be 10 steps long and the two triangles together total 20, just about right on the box that measures 30 on each side. They make sure the pen is up while the turtle moves to the starting point for the bow, then they put the pen back down and watch the turtle robot draw out the bow they have designed. Wow, now they have a beautifully wrapped present for Val!

What is going on in this classroom? -- learning by discovery, experimentation, and manipulation. The children are learning LOGO, but their first introduction to this powerful computer language is through a concrete object they can see and touch -- a turtle robot.

Val is the class name for the Valiant Turtle Robot. Children need actual experience with concrete objects, especially at the kindergarten and early elementary levels. These manipulative activities with the turtle robot allow the children to understand the LOGO commands and what they cause the turtle to do. When they have the opportunity to be at the computer themselves, the small triangle on the computer screen is obviously a symbolic representation for the turtle robot they have observed in class. They recognize that they can maneuver the graphic "turtle" on the screen the way they have done the turtle robot on the floor.

Val, the turtle robot, looks like a turtle with eyes, legs, and a tail. But it also looks like a robot. Through the shell the children can see the computer card, the chips, and the wires which receive the signals from the communicator box and make it move. They can even experiment with blocking the communication signal and come to understand that Val cannot do anything unless it receives the command to do it. Val can also demonstrate very artistic designs. With its pen down on the paper, the turtle can draw whatever the children command it to draw. They can acquire knowledge about angles and shapes by drawing fun objects with Val.

Val is an excellent way to introduce children to LOGO. The students learn the basic LOGO commands as a group. The turtle robot activity prepares the children so their actual computer time is positive and purposeful.

Students also learn problem solving skills in a non-threatening group setting. They practice making predictions and testing the various alternatives. They see how to revise their predictions along the way and how to seek new alternatives toward the solution. Because they have reasoned through the robot's movements first as a group, the children have greater confidence and freedom to explore the LOGO possibilities once they come to the computer to work independently.

The learning which has taken place with Val, the turtle robot, in the classroom transfers easily to the small triangle "turtle" they see on the screen. They can clearly visualize the number of turtle steps Val needs to travel. They begin to understand the degree of a turn needed to tell Val how to get where it needs to go. The students can use the knowledge they acquired during this discovery period to experiment on their own. Because they have manipulated the robot, they can confidently approach problem solving with the turtle character on the screen.

Discovery learning through manipulation is essential to the understanding of the nature of LOGO. This can be accomplished in the classroom by formulating a series of commands for a student (using a stuffed toy) to follow, but it is so much more effective with the turtle robot. The children are fascinated that this turtle cannot do anything without their instructions. They are amazed when the turtle responds to their commands and they learn to be precise in these commands so that the turtle does what they want it to do.

Discovery learning and manipulation come together perfectly in the Valiant turtle robot. Using the robot in the classroom can help introduce the basic LOGO concepts, can stimulate the creativity and experimentation necessary to successful problem solving, and can result in the full integration of the computer into the curriculum.

CROSS OVER THE BRIDGE - A JOURNEY TO ABSTRACTION WITH LOGO AND MANIPULATIVES

JUDITH OLSON
WESTERN ILLINOIS UNIVERSITY
MACOMB, ILLINOIS

PATRICIA LANG
FAIRVIEW ELEMENTARY SCHOOL
CHEYENNE, WYOMING

The passage from concrete to abstract development of concepts is not so much a bridge as it is an entire journey. The process of abstracting grows through many experiences at different levels of concreteness and abstractness. Since elementary school age children are in the process of acquiring the cognitive skills necessary to handle more abstract mathematical concepts, we need to provide a mechanism to bridge concrete to abstract thinking. Piaget emphasized the need for physical experience in the development of mental action. He divided experiences into two parts, physical and logical-mathematical, which contribute to the formation of structures in the mind that children call upon in the future. The use of manipulatives in concrete experiences, which can be visually represented with Logo, is one way to enhance the development of abstract mathematical concepts. Through the combined use of manipulatives and Logo, explicit linking of different representations of mathematical concepts can make the journey to abstraction exciting. Understandings that are developed from manipulatives are translated to the computer screen via Logo. Appropriate questions are necessary to direct the students' attention to mathematical ideas and to prompt further explorations. However, we must restrain from expecting formal conclusions from young students. They must be given time to investigate, to ponder, and to explore ideas of their own making prior to discussing their discoveries with each other.

In any elementary school classroom, children are likely at several different levels of cognitive development. Therefore, concepts also need to be presented at different levels. We know students can benefit from exposure to multiple representations of the same concept. Research in elementary school mathematics (Driscoll, 1981) indicates that young children are predisposed to spatial modes of learning and that their spatial perceptions strongly influence their mathematical conceptualization. For this reason, teachers are encouraged to use the visual mode of instruction to help children on their journey from concrete to abstract thinking. However, if the visual mode

is used alone, relying on children's visual perceptions without concrete referents, problems may occur.

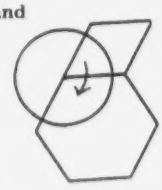
Clements (1984, 1985) has been involved in studying the effects of Logo programming on cognition. He studied the delayed effects of computer programming in Logo on the cognitive abilities and mathematics achievement of primary grade children. Children who had received Logo instruction or CAI as first graders were tested and interviewed in their third grade year. As some children in the Logo group were interviewed, they drew a single, slanted line segment when asked to "draw an angle". Other misconceptions included angles of poloygons being conceived as sides and the number of angles in a triangle being two or four because the children related them to the turns of the turtle. This illustrates how a child's work in Logo may remain disconnected with other aspects of his or her mathematical knowledge. The interviews pointed out lack of conections between what is done in Logo and the concepts developed in the mathematics classroom. Here the use of manipulatives serve a as valuable instructional resource to make the concepts come alive while also providing an explicit awareness of the mathematical concepts involved in the use of Logo. Teachers must consciously design learning situations so that students see the connections.

From manipulating objects, students can identify the relevant attributes from which definitions are made. If the experiences are to provide enhancement of geometric concepts, activities with Pattern Blocks, Tangrams, and other manipulatives must precede the corresponding visual and abstract activities. Our presentation will include activities that have been used successfully with teachers in inservice classes and also activities specifically used with fourth, fifth, and sixth graders in a Logo Lab. One of the first things we have students do, after they have learned how to write procedures and manage files, is to build a file of procedures for a set of Pattern Blocks. While this can be done without variables at first, the use of variables increases flexibility.

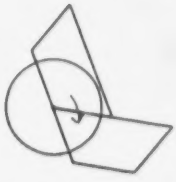
After students have experimented with Pattern Blocks to discover relationships, an activity such as the following, adapted from Let's Pattern Block It (McClean, Jenkins, and McLaughlin, 1973), should precede students writing procedures for the Pattern Blocks.

In a Logo environment, the turtle must turn the external angle to draw the next side of a polygon. The concept of internal angle and external angle can be developed using Pattern Blocks and straight lines. The hexagon and the small angle of the rhombus put together form a straight line or 180°. The internal (inside) angle (turn) of the rhombus is equal to the external (outside) angle (turn) of the hexagon.

What is the measure of the internal angle of the hexagon?
What is the measure of the external angle?
What about the measures of the internal and external angles of the rhombus?



Try the same procedure with the two different angles of a trapezoid to see the turns of the turtle necessary to go around the outside of the trapezoid.



By manipulating the Pattern Blocks in this way, students will have opportunities to internalize angles as turns and apply the concept of internal and external angles in a concrete way before using the visual mode of Logo.

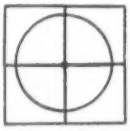
Olson/Lang 3

SAMPLE ACTIVITY

A circle is measured in units called degrees. When we measure the numbers of degrees in a circle, we find it always has 360°.

Let's explore Pattern Block pieces to discover how they fit on a circle. This will help us find the number of degrees in the angles of the blocks.

How many squares cover a circle?

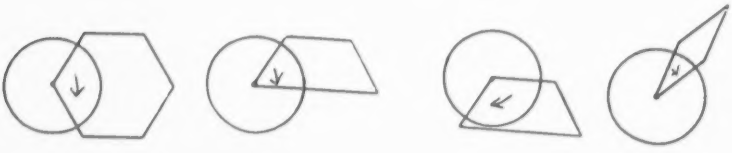


Since it takes four squares to cover the circle and there are 360° in the circle, there are 1/4 of 360° in one angle or corner of one square.

How many triangles cover the circle?

Each angle measures 1/6 of 360° or 60°.

Try this procedure with the hexagon, both angles of the trapezoid, the rhombus, and the kite to find the number of degrees in the angles of each of the Pattern Block pieces.



Why should we use Logo for the visual mode in developing these mathematical concepts? We feel Logo provides a thinking and problem solving environment in which the students actively participate in the conceptual formation of mathematical ideas. After thinking about and talking about what they are doing, the abstract concepts should be internalized and easily applied in new contexts by the students.

Participants will be involved in similar activities involving other mathematical manipulatives such as miras, tangrams, and geoboards. Our presentation will conclude with a short video showing fourth, fifth, and sixth graders at Fairview Elementary School at work in the Logo Lab where many of these ideas have been successfullly used.

REFERENCES

Clements, Douglas H. "Longitudinal Study of the Effects of Logo Programming on Cognitive Abilities and Math Knowledge". Paper presented at annual meeting of NCTM, April, 1986.

Clements, Douglas H. and Dominic F. Gullo. "Effects of Computer Programming on Young Children's Cognition." Journal of Educational Psychology, Vol. 76, No.6, 1984.

Driscoll, Mark J., Research Within Research Elementary School Mathematics. St. Louis: CEMREL, 1981.

Ginsburg, Herbert and Sylvia Opper. Piaget's Theory of Intellectual Development. New Jersey: Prentice-Hall, 1969.

McLean, Peggy, Lee Jenkins and Jack McLaughlin. Let's Pattern Block It. Hayward, CA: Activity Resources, 1973.

Martin, Kathleen and Donna Bearden. Mathematics and Logo. Reston: Reston Publishers, 1985.

An Environment for Action

Rebecca L. Poplin
Teacher, Barwise Junior High, Wichita Falls, Texas
Logo Exchange Columnist

"Environments for action" are everywhere. A football stadium is one. Action certainly occurs, but most of the people are watching rather than creating it. A beehive is an environment where action abounds. Activity is organized; each bee has a certain role to play. But the bees' routines are highly structured, varying little. A social gathering is an environment for action. Activity is mostly random. Roles are generally undefined. The success of the venture is subject to emotion. None of these examples approach a model for a Logo environment. A Logo environment is alive with action. It feeds on interaction. It grows and changes every day. It needs to reproduce to thrive.

Action - Living

Just as every living thing is unique, so is each Logo environment. Many have tried to explain, quantify, and teach about "it", but there is no one way to describe a Logo environment. At best we can only share our ideas of the best way to use Logo or discuss what we have experienced. There are probably some underlying truths that are universal, but Logo environments are just too intricate to be explained simplistically. The complexities of each situation are the ingredients which make a learning/teaching environment exciting. Life is never the same from week to week. Every day has new ingredients. Each hour holds new potential.

A Logo environment is infused with life. It almost breathes and pulses with its own rhythm. There is activity everywhere, both above and below the surface. Ideas are popping out all the time. Learning jumps from person to person like impulses travel in the nervous system. The Logo underground is the circulatory system.

"Wow, that's really neat!" "How did you do it?"
"I want to know how to make the computer do music like Leah did."
Teacher: "How did you figure this out?"
Student: "Well, Brian showed me this neat design, so I changed it and this is what happened."

Students take so much control over their own learning that some teachers have a problem. They cannot claim to be in charge or realistically take credit for "teaching" what the students are learning. The teacher is no longer the brain or even the nerve center of the classroom.

In the living Logo environment, the teacher may have many roles, but the primary one is that of providing nourishment. Most students experience Logo burnout from time to time. Some feel frustrated with what they are able to do compared to other students. Others have been trying and trying to do a particular thing and finding it too hard to do alone. Still others will finish a task and have no ideas about what to do next with the computer. Without the proper nourishment, the burnouts can lead to turnoffs. A very sensitive teacher needs to determine just the right way to refuel the students so they can and will try again.

Life is not a problem to be solved but a mystery to be lived. Logo is not a problem to be solved, a language to be taught, or an experiment to be validated. Logo is more of a mystery for which each character writes a unique plot.

Action - Interaction

A living learning environment thrives on interaction. In the same way, learning with Logo seems to need the interaction of others to develop and grow. How many students with computers have Logo at home? How many of those use it regularly on their own in preference to video games? How many students who use Logo regularly at home have no support system at school? Is it realistic to believe that students from 5 to 18 will ever use Logo extensively without this support? Interaction with a creative enthusiastic teacher and with other students who are also involved seems to be a vital ingredient for success.

Just as students need mutual support, so do teachers. Logo users' groups, newsletters, conferences, and networks must be a part of any teacher's support system. Not every teacher in a building is likely to use Logo, especially at first. In fact, very few schools have a majority of teachers using Logo. Of these, the successful schools have extensive administrative support systems in which principals and/or coordinators take an active role in the process and encourage the teachers to grow and support each other.

As author of "Teacher Feature" for the Logo Exchange Newsletter, I have the opportunity of talking with teachers all across the country about their Logo experiences. They represent a broad spectrum of interests, projects, and philosophies, but they have in common an enthusiasm for learning and a desire to talk about what they are doing. We need to get together. We need to share and support each other. We need to stop speaking loftily about what the ideal "Logo Environment" is and talk more about what we have really learned from our own experiences.

Action - A Continuum of Growth and Change

One of the most exciting things about living in the Age of Information is the speed with which the picture changes. Computer technology is changing so fast that most of us cannot keep up. Just a year or two ago, it was possible to know almost everything about one type of computer. Now that goal is futile. Logo is changing. Logo changes people. Logo users change in their understanding of it.

Our relationship with Logo can be like our relationships with other human beings. In our relationships with others we grow closer; we grow further apart. We understand each other better; we are amazed by each other's unpredictability. Whether we stay together depends on the strength of our interdependence.

Every teacher I have interviewed has changed while using Logo. Many of them credit Logo with opening doors of understanding, with confirming nebulous ideas about learning, and with showing them how learning can happen. Several teachers have reported that they used to teach Logo as a language, but that now they are doing more specific and focused things - focused on ideas that are interesting to learn or on topics that arise from other studies. Logo is becoming another way to express concepts and to understand reality.

Moving from a somewhat laissez-faire approach to a more cooperative, supportive, and nurturing attitude is a typical change that teachers experience. Teachers who used only Logo in the beginning of their computing careers have embraced other computing tools. Word processing, graphics, and newspaper software have become staples on the classroom computer shelf. No apology is needed for using the computer without Logo. As technology expands, more tools are available to enrich our lives. We must not become myopic because of our enthusiasm about what Logo will do. It is a springboard and a model, but it is not the solution to a problem.

Action - Explosion

"How can Logo survive?" is a question frequently heard from panelists and conference participants. Logo is alive and well in many places, but as more and more teachers begin using Logo, there are more and more chances to lose sight of its purpose. In order for Logo to survive there must be an explosion.

First, we must explode territorial lines which separate us from each other. We all face the danger of becoming self-righteous about our view of what Logo should be. We have to open the doors and accept each other at face value while sharing points of view. In individual settings, the computer boundaries must disappear. Having computers clustered together in a separate lab where one or two brave teachers venture to do all and be all in computer education seems like the proverbial library where the ideal state is to have all the books one inch from the edge of the shelves and dusted.

There are lots of outstanding Logo teachers in this country. But too often I ask one of them, "What about the other teachers in your school? Are they involved in Logo?," and the hemming and hawing begin. Most say "Weeeelll, a few are interested in the possibilities, but most of them are not interested or too busy. Computer use must explode within schools and districts.

Administrative support is vital. School principals cannot sit back and watch the computer age happen without them. Each school needs a strong leader who is interested in providing the best possible learning environment for children and for teachers. Where that attitude exists, growth and change must occur. Every teacher who uses a computer has a responsibility to share that knowledge and encourage other teachers to learn skills which will make time more effective. Where the principals sets the stage for cooperation, this kind of explosion is more probable.

We need an explosion of professional enthusiasm. As educators, what we can and must hope is that the profession will attract the brightest and best young people to participate in the learning community. Then Logo needs to be present, available, and understandable - not just as a computer language but as an avenue to thinking and learning. Learning is learning how to think. Learning happens because of involvement not bombardment. If Logo is one way to involve people in the act of learning, then professional educators must keep it alive.

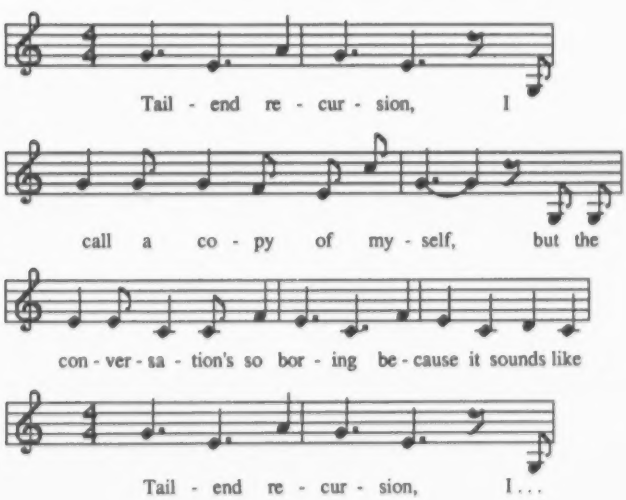
BUDGET-MINDEDSTORMS

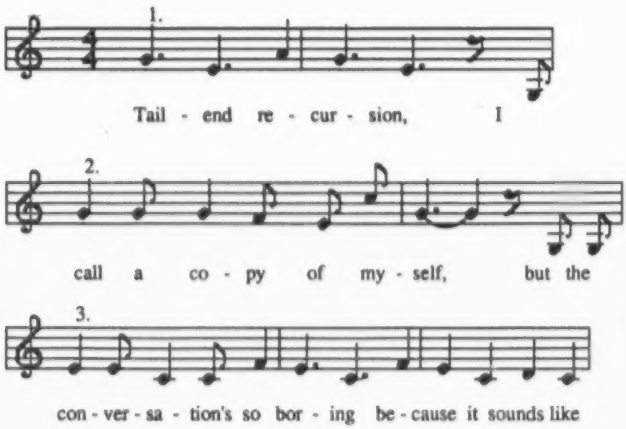
© 1987 Peter Rawitsch

Peter Rawitsch
Glenmont Elementary School
Route 9W
Glenmont, NY 12077

If superprocedures had to pay a service charge each time they called a subprocedure, the monthly bills that I would run up would be astronomical. Fortunately, the cost would be justified by the quality of the service. MIT Logo, for example, provides several custom calling services for my convenience. "Touch-tone service" is available with the READCHARACTER command and "call forwarding" is possible with IF and THEN. A "phone book" is always handy and up-to-date with POTS. The biggest benefit of these services is that I can have many and varied procedural conversations.

For the budget-minded procedure caller, there is recursion. This special service gives you the ability to call a copy of yourself. The rationale for why you need this service may not be apparent at this time, but the price is affordable. The reason it's so cheap is that you only get a *copy* of yourself, not the original. (Multiple originals are available when you install a REPEAT command.) The main disadvantage of this service is that the conversations are boring. They sound just like the original. Here is a musical model of what the beginning of a tail-end recursive conversation sounds like:

```
TO MYSELF
   PR [Tail-end recursion,]
   PR [I call a copy of myself,]
   PR [but the conversation's so boring]
   PR [because it sounds like]
   MYSELF
END
```

This model goes on forever. In a classroom, children can stand in a line to sing MYSELF recursively. The first child sings MYSELF and calls the second child (who is a *copy* of MYSELF.) The second child sings MYSELF and calls the next child (who is a *copy* of the second MYSELF) and so on. Having the children use paper cup and string telephones to call their copy might help some of them make a better connection as to what's happening in the recursion process. In each case, the END is never reached in the procedure.

The musical name for recursion is a *round*. MYSELF can be sung as a two- or three-part round. Small groups of children begin singing it at the places marked by numbers in the score below. Each group would sing MYSELF and a *copy* of MYSELF and a *copy* of a *copy* of MYSELF ad infinitum.

<CONTROL> G is one method of stopping tail-end recursion on the computer. Designating one child to hold up a prepared "<CONTROL> G" sign when you tire of singing or listening is suggested.

I don't have a need to call MYSELF (and its copies) too often. I use this musical model to introduce the idea of recursion to Logo students. It also offers some simple harmonies and interesting rhythm combinations. (It may be of some interest to Logo musicians that all seven measures of the round must be sung before the harmony and rhythm patterns repeat.)

I am a satisfied customer of the Logo procedure system. It has the latest modular equipment, efficient mathematical operators and helpful messages when I press the wrong buttons. However, what I am most grateful for is that Logo's procedures are toll-free.

"Adventuring in Logo"

by Curtis Ring
Wooster High School Wooster, Ohio

Commercial adventure games are of great interest to young people. Many students who enter programming classes would like to write their own adventure games. Writing an adventure game is an excellent vehicle for teaching a number of computer science concepts: recursion, list processing, a little bit of artificial intelligence, and comparing when to use local variables as opposed to global variables.

Proper use of recursion is needed to keep adventure games from not ending. A stop rule such as the player being "killed" or the player successfully completing the game must be applied in the adventure game's code. The user moves from one place to another within the game. The program handles this process by executing different pathways throughout the game's code. However, the program must always return to the user for a command.

List processing is crucial to Logo adventure games. Procedures that allow the user to command the adventure with English word structures that resemble actual sentences must accomplish the task of finding parts of speech within these commands.

The simplest of these commands will be a verb and a noun (i.e. "get lantern"). The procedure that accepts commands from the keyboard must separate the noun from the verb. The verb will have to tested to see if it is part of the game's vocabulary, as will the noun. If the verb is part of the game's known words, a procedure with the same name as the verb will be called. The result of these verb-procedures depends upon the verb entered, most will refer to moving the player in certain directions or picking up or dropping the nouns of the game's vocabulary.

Specific tasks like "get"ting and "drop"ping game objects are classic examples of searching for and adding or removing elements in lists. The player's "inventory" is a list of all the items the player possesses. More specifically, it is a list of lists as the sub-lists may contain a decriptive name for a game object (such as [gold key]

or [little black hat]) A "get" procedure adds game objects to the inventory, which isn't very difficult The hard part is the procedure that drops objects from the player's inventory A "drop" procedure must search through the inventory list until it finds a matching sublist, removes it, and then puts the inventory list back together again A task that is easier said than done

All adventure games must have "parsers" Parsers convert the user's commands to actual procedure calls Logo's list operators make designing a parser that takes nouns, verbs, adjectives, determiners, and some conjunctions as input not only feasable but manageable Logo parsers have master lists that contain all of the game's vocabulary, one each for verbs, nouns, adjectives, and conjunctions These master lists may also contain synonyms for other words, for instance "lantern" may be a vocabulary word and "lamp" can also be in the noun master list, even though both words refer to the same game object

Adjectives and conjunctions are the most difficult parts of a Logo parser to create They are very special cases since not all of the user's commands contain these parts of speech They invoke procedures that make special use of both nouns and verbs "And" usually involves a compound command structure, the player wishes to do two things with one command (i e "get the lamp and go north") Conjunctions usually involve the use of nouns in a selective way The player will want to unlock a door with a silver key and not a gold one The parser must find these conjunctions and execute the corresponding verb procedure using the noun specified

Adventure games, by nature, must make appropriate use of global variables The player's location within the game, for instance, must be global because so many different procedures make use of this value All the procedures that move the user from room to room must have the location value Any procedures that invoke game "specials", unique and surprise happenings in a certain

place within the game, also require the location of the player The procedure that tests for the player's successful completion of the game must also have this value, as some games depend upon the player reaching a destination within the game in order to win

Logo, as a programming language, is a unique tool for creating adventure games Logo is a list processing language and so lends itself to working with words more than programming languages like BASIC and FORTRAN that rely heavily on numbers In the case of BASIC, Logo's procedural structure along with its ability to do recursion make it the clear choice for programming adventure games Logo's local variables make progams simpler to debug than any programming language that uses only global variables Logo also bests even PASCAL with its ability to handle words Logo code is easier to read than that of PASCAL Logo also invites experimentation and exploration with its editing capabilities and its interpreter With students, PASCAL tends to dampen initiative because of its stringent syntax rules PASCAL's file structure, while very complete, is not as manageable as Logo's

Logo can prompt student interest in programming The adventure game is one way to spur this interest and, at the same time, teach some important computer science topics Adventure games show proper use of recursion, a little bit of artificial intelligence, list processing, and when to use local and global variables

Exploring the Algorithms of Written English

The purpose of this presentation is to discuss the use of the microcomputer to explore the conventions and rules of written English. Learning to read and write is a task that millions of students begin each year and continue for several years in early elementary school. For many of these children their first encounter with written text is one of natural interest and curiosity. This may soon be replaced with frustration as text is decomposed into its various parts to provide children with lessons about written language as we know it. Much of this instruction involves teaching children rules and exposing them to convertions organized to create learning experiences that will further their literacy.

Unfortunately, some of these teaching techniques are problematic to young learners. Too often the lessons do not put to good use many of children's fundamental learning strategies. Rather than encouraging children's desire for exploration or examination, they require chilren to memorize what they have been taught. For example, students are asked to memorize a rule that states "ea" makes a long "e" sound. That works fine for some words but is not useful in spelling many words like bread, thread, dead, lead, etc. The truth is many that of these rules have a limited use which is often not described or explained.

We have found significant advantages in teaching literacy with the use of microcomputer software that provides an environment in which children can explore written English and make sense out of its many convenions and rules by constructing their own text. This is accomplished with the use of a "talking word processor" that gives students control over what they construct and how hey alter those constructions. This "text-talkin construction set" lets the learner discover some of the aspets of written English that even "confuse" a computer. By privіding children with a set of tools to test out what they think is working and modify whatever they reject, this talkin textwriter lets children draw their own conclusions about written English. Such an approach sustains children's interest and offers them a more accurate understanding of how written English works.

Teresa J. Rosegrant Ph.D.
Associate Professor
George Washington University
School of Education and Human Devdopment
Washington D.C., 22052

Logo in the Microworld of Life

Brian Silverman
Vice President, Research
Logo Computer Systems, Inc.
9960 Côte de Liesse Road
Lachine, Québec, Canada H8T 1A1

Thinking of Logo conjures up images of kids and turtles. This presentation will be about a Logo microworld which doesn't have any turtles and isn't for kids.

People have often tried to come up with a definition of a "microworld." I don't think that any definition has come up yet that is compelling enough to be adopted. Presenting a microworld that is very different that the one that we are used to can shed some light on what we really mean by "microworld."

The Life Game has been popular amongst computer enthusiasts for many years. In it the computer screen is divided up into an array of square cells. Each cell is either alive or dead. A starting position is set up by the player by arranging a pattern of live cells. Every move, or generation, is made by having the computer apply a couple of simple rules to all of the cells. The rules trigger a number of births and deaths that change the shapes and sizes of the colonies.

The evolution that occurs over numbers of generations is quite fascinating and surprising. Over the years people have catalogued some of the flora and fauna that emerge in Life.

A few years back, I started playing with Life. As a Logo enthusiast could be expected to do, I considered the ways of using Logo to help in the exploration. What resulted was a Life Logo. This is a Logo in which the turtle graphics microworld has been removed and replaced by the Life game. There are about a dozen new primitives that control the setting up of an initial position and the flow of generations.

The exploration of Life kept me going for about a year. There were a lot of experiments to be tried to get a handle on what was happening. None of them led to much insight. What appeared at first to be nearly but not quite understandable turned out after careful study to be nearly but not quite understandable.

After that first year I figured that there was no point in being bound by the rules of Life. I adapted Life Logo to be sufficiently flexible so as to allow for changes in the rules.

Different sets of rules led to dramatically different activities. One set of rules led to electronic circuit design, another to reconsidering Darwin, and yet another to some obscure topics in number theory.

Life Logo is a "microworld." Without even trying to pin down the word "microworld," it's possible to try to characterize some its properties. Both Life Logo and turtle graphics provide for the possibility to explore a wide variety of different topics. In both cases the software doesn't provide any direction. It just provides a context. What's interesting is that the topics that emerge are not topics that are "in" the microworld. Making squares can lead to thinking about angles. Playing with Life can lead to thinking about evolution.

For me, exploring Life was another case where Logo proved to be one of the easiest ways in to a complicated problem.

Logo Thoughts: LogoWriter and Logo II

Brian Silverman
Vice President, Research
and
Michael Tempel
Director of Educational Services

Logo Computer Systems Inc.

Brian's thoughts:

LogoWriter and Logo II were both produced by LCSI and both run on Apple II's. LogoWriter is two years younger. Usually, that sort of information would be enough to jump to the conclusion that LogoWriter has made Logo II obsolete. There is, however, room in the world for both of them.

There are two distinct uses of computers in education that often get confused because the border line between them isn't very clear. First, computers are something to learn about. Second, computers are something to learn with. Until LogoWriter, Logo had always been in the grey area between the two.

Learning about computers is learning about programming, data structures, modularity, syntax, control structures and all the other topics that would round out a computer science curriculum. It is a subject necessary for anybody wanting to pursue a career in computers.

Learning with computers is like learning without computers except that computers have been added. It could be about any subject from art to science to language to anything at all. The computer is there to help out just as other materials had been there to help out in the past. Of course to learn with computers, you'd have already had to learn a bit about computers, but the level of knowledge and sophistication needed is nowhere near what you'd expect from a computer science major.

Logo II is ideal for those who want to learn about computers. It is a complete programming language with plenty of features and lots of user memory. It has file handling, error trapping, property lists, output formatting, and just about anything you'd need to launch into any computer science project.

LogoWriter, on the other hand, is admittedly lacking for those who want to learn about computers. Its design was biased to make it appropriate for those who want to learn with them. There is more emphasis on graphics and text manipulation and the user interface has been streamlined to hide much of the inner workings of memory and file management. Thus LogoWriter is more appropriate for the arts, language, science, or math class.

Logo is just a language. A language can be used to express many things. Logo II and LogoWriter share the same language. They differ in their vocabularies. LogoWriter is especially good at talking about text and graphics. Logo II leans more towards computer science.

Michael's thoughts:

Brian is correct that there is a place for both Logo II and LogoWriter, but the division may not be between learning with computers and learning about computers.

Many of the features of LogoWriter are clearly advantageous to someone learning about programming. No Logo II programmer would object to being able to edit and re-execute a series of commands at top level and then copy them directly into a procedure. The ability to format procedures and editing features such as CUT, COPY, PASTE, SEARCH, and REPLACE make program development easier regardless of whether you are learning about the program itself or using the program to explore some other subject area.

The richness of the LogoWriter environment provides more ways of working with a particular idea in computer science. If you are working with lists, you have lists of multiple turtles and shapes as well as lists of words to works with.

The limitations of the current version of LogoWriter that result from squeezing it into a 64K machine make it less suitable for writing large complex programs. Memory is limited and there are a number of primitives found in Logo II that are not in LogoWriter. While replacements can generally be written in Logo, they often run more slowly than primitives and use up some of the already scarce memory. These shortcomings may make the original LogoWriter unsuitable for advanced computer science courses, but do not necessarily make LogoWriter less appropriate for learning about computers, at least at the introductory level.

"If There Had Been Logo... I Might Have Learned Math"

Gary S. Stager
Network for Action in Microcomputer Education

"You can lead a child to Euclid but you can't make him think"
-Seymour Papert

Whenever I attend a Logo conference or conduct Logo teacher training I hear countless revelations from Logo users about what lousy math students they were. These adults commonly admit that they previously disliked mathematics and that their use of Logo has created a new sense of interest and purpose for learning mathematics. I must confess that I was a very poor math student. Despite my programming ability, I steered away from majoring in computer science because of my distasteful and ineffective mathematics education. It is my belief that our schools have systematically created several generations of citizens who are math phobic or mathematically illiterate.

Seymour Papert would suggest that many of us were "mathematically alienated". We were alienated because mathematics is abstract and we were expected to learn it through drill, instead of being immersed in a culture where we could learn mathematics in a natural, more intuitive and concrete manner. A small percentage of students do learn mathematics. This is because their individual learning styles are compatible with the way mathematics has been traditionally taught. Most of us were not so fortunate; our learning styles were often in direct conflict with the way in which mathematical concepts were presented. We had other people's thinking imposed on us with we were rarely made aware of alternative ways of solving the problem at hand. One learning style has been imposed on the entire population. It would appear that much of mathematics education is unaware of the work of cognitive psychologists and epistemologists.

If we did learn mathematics, we didn't internalize the knowledge and have long forgotten what we "learned." Mathematics was presented as something that was hard, cold, elitist, and externalized from our own experiences. As students, we saw little meaning or practical applications for learning math. Since we have discovered Logo, many of us have developed a new sense of interest and purpose for learning mathematics. Logo has allowed countless adults and students to experience the MAGIC of mathematics.

Of course, I am making sweeping generalizations to illustrate my point, but I find an alarming amount of educators who's self-esteem has been bruised by their perceived inability to "do math." When I teach a Logo workshop for teachers I am often confronted with, "Oh yeah, you can do that, but you can't possibly expect us to be able to do it." This phenomena occurs even with the simplest of mathematical ideas, such as drawing a circle or measuring a diameter. I'm often amused by this perception that I am a mathematician (I have a box full of report cards that would provide strong evidence to the contrary). I then slowly break the problem at hand down into "mind sized chunks" that the person can relate to because of past experience with those ideas, then through the use of Logo and some of its powerful metaphors the problem becomes understandable.

In my opinion, traditional math curriculums make a tragic error by presenting math in a

© 1986 - Gary S. Stager

rigid sequential fashion. First you answer the problems on the page, then you do the next page for homework; eventually you do the chapter review and take a test on what you have learned. We now know that knowledge is not only acquired sequentially, but also in fragments that have many applications and might eventually be "glued" together. This page 1, page 2, chapter 1, chapter 2... way of teaching/learning is not natural and doesn't lead to an understanding of why we should bother learning anything at all. Other subject areas are guilty of this type of instruction, but not to the extent of mathematics.

I believe in "seizing the teachable moment." If a third grader needs to use decimals to solve a problem, I provide him with enough knowledge to tackle the problem. despite I do not concern myself with the fact that decimals are not taught until the fourth or fifth grade. The third grader doesn't understand everything about decimals, but when the concept is formally introduced he will have previous experience with "decimal-ness."

The ideas I have just presented give me an image of a teaching/learning tool that currently exists in other disciplines, but rarely in mathematics instruction. This image is that of a laboratory where knowledge transcends the page and you can touch it, and spill it, and mix it. There are math learning centers in today's elementary classrooms, but unfortunately as move further away from the heyday of open education, the presence of math centers and tactile manipulatives is diminishing. In my presentation I am going to demonstrate ways in which Logo can become a powerful math lab.

Logo provides the student with an educational microworld in which they can explore and manipulate many abstract mathematical concepts. The child can "mess about" with this newly discovered knowledge and in a short time realize that these simple, powerful ideas can be combined to produce complexity. With Logo, the child becomes deeply involved in his/her learning process and develops a learning style. The student's curiosity and new-found interest will enable him/her to discover new concepts by the manipulating what he/she is already familiar with .

Everybody knows that turtle graphics has many educational benefits for the child. By teaching the turtle to draw a figure, the student must assume the role of the turtle in his/her own space and reflect on how they themselves would draw the figure. This process encourages problem-solving and reinforces mathematical concepts inherent in the particular programming activity. The child takes control of their own learning and establishes a personal relationship with mathematical ideas.

Turtle geometry provides a rich "mathland" in which movement, measurement, turning, arithmetic operations, logic, and many other mathematically powerful ideas can be explored. Students programming with turtle graphics are bound to gain an understanding of numerous mathematical issues.

The Presentation:

Although, turtle geometry has infinite educational possibilities, there are other areas of the mathematics curriculum that can be addressed by Logo. In this presentation, I intend to show several Logo math microworlds that I have developed. I am indebted to E. Paul Goldenberg. Paul's notion of procedural machines and iconic notation is so logical and powerful in its simplicity that it pervades the way in which I now think about Logo and mathematics. This powerful idea is contained in many of the Logo examples listed below. These Logo procedures attempt to address issues dealt with in the (K-8) elementary school

mathematics curriculum and lay the foundation for more advanced concepts. The examples I demonstrate will be in LogoWriter and they include:

Fractions: Microcomputers do not generally deal with fractions, although this is a consistent source of difficulty for many students. I will demonstrate "fraction machines" or "fraction calculators" that allow you to manipulate numerators and denominators, add, subtract, multiply, and divide fractions, and reduce fractions to their lowest terms. There are also procedures for simplifying mixed numerals and fraction-decimal conversions.

New Math Operations: When you attempt to solve mathematical problems, in Logo, you realize that there are mathematical operations that need to be added to Logo. Some of these operations exist in other versions of Logo, but I wrote them for LogoWriter and Apple Logo I . Many of these procedures are simple to create and by doing so students gain an understanding of the particular mathematical operation. Many of the following new math operations are of particular use to students in the elementary school: Absolute value (ABS), the distance between any two points (Distance), the ability to use exponents (POWER), division of real numbers (DIVIDE), (DIFFERENCE) between two numbers, and a procedure that lists all of the factors of a number (FACTORS).

New Math Predicates: Logo predicates test a condition and return a value of true or false. They are of paramount importance and many new ones are useful for understanding mathematical conditions and in programming projects. Some of the ones I have created include: POSITIVEP, NEGATIVEP, ODDP, EVENP, PRIMEP, COMPOSITEP, INTP (is the value an integer?), and FACTORP (is one value a factor of another value?).

Baby Turtles: I will demonstrate very simple and effective ways to customize the LogoWriter (or Logo) environment for beginning Logo users in the primary grades. This microworld provides easy transfer of ideas to a complete Logo/turtle environment without the added confusion of inferior commercial "baby Logo" packages.

Counting Machines: These are procedures that allow the child to ask the computer to count up or down, between any two numbers, by any increment or decrement. This is useful for learning addition, subtraction, and counting.

The Perimeter Tracer: The perimeter tracer is smarter than the average turtle. As this turtle draws, it keeps track of how many steps it has taken. Very useful for learning distance, perimeter, and area.

The Pie Chart Procedures: These procedures will take a list of numbers, either inputted by the user or from another procedure, and draws a pie chart with each "slice" in a different color. The percentage of each "slice" to the whole will be noted on the screen. I will also share simple bar graph and line graph tools.

The Symmetry Toys: These procedures allow the student to use multiple turtles (in LogoWriter or Sprite Logo) and explore various aspects of symmetry. I also have procedures for traditional versions of Logo that simulate multiple turtles by drawing reflections of the turtle's movement in two-four other quadrants. All of these "toys" provide a microworld for "messing about" with symmetry.

During the course of this presentation I may share with you many other Logo math tools that can be used by the child to explore: the cartesian coordinate system, positive and negative numbers, decimals, probability and statistics, similarity, pi, circumference, radius, diameter, the total turtle trip theorem, and many other concepts. I have created these math tools for and with my students, and I will present examples of their work.

I have also explored Seymour Papert's ideas of "thinking mathematically" and "mathematics as a language". As a result of these ideas, I have created a microworld in which primary grade students can express mathematical ideas in their own words. A first grader might say, "How much is twice 3 more than 5 takeaway 2?" The computer would then return an answer and might even ask the child to solve a similar problem.

Older students may also solve problems in a similar fashion. The notion of simple procedural machines that take a value as input, operate on that value, and output a new value is a very powerful idea. A student could create a series of one and two line procedures that could be used to solve several different problems, for example: "How many 3.5 inch thick books will fit on a 7 foot shelf?" As you can see, the problem is expressed in "mathematical language" and must convert 7 feet to inches and divide 3.5 inches into 7 feet.

Thinking mathematically is not exclusive to solving mathematical problems, but has applications in all subject areas. LogoWriter provides a rich environment in which mathematical thinking can be transferred to other domains, such as, art, music, social studies, science. I'll share ideas for encouraging this type of thinking across the curriculum.

Conclusion:

The power of all of these Logo math tools and microworlds is that they are all flexible and may be used in conjunction with any other types of computation. One might suggest that they be thought of as mathematical building blocks. This truly provides the student with a microworld in which simple powerful ideas are capable of being developed into much more complex concepts. An early successful Logo experience can boost a student's self-esteem and change the way in which he/she thinks about her/himself, school, and learning. This new found sense of accomplishment and thirst for knowledge can significantly brighten a school career that may otherwise be destined for failure and disappointment.

It is my hope that my thoughts, anecdotes, and tools will create interest in mathematics and stimulate the child's imagination and curiosity. I hope that I will also challenge teachers to reflect on and re-examine the way in which they think about mathematics and teaching mathematics. Maybe the next generation of boys and girls will be turned on to mathematics and will perceive it as meaningful, challenging, and powerful. Logo can play an important role in the democratization of knowledge. Mathematics is something that can be learned and enjoyed by children, possibly avoiding another generation of "mathematically alienated adults". I believe that Logo can significantly help us towards this goal.

Bibliography

Harvey, Brian. Computer Science Logo Style: Intermediate Programming. Cambridge: MIT Press., 1985.

Goldenberg, E. Paul. "Building Mathematical Systems: Using OUTPUT to Build Functions from Elementary Word Problems Though High School Math." Logo '86 Proceedings. Cambridge: MIT Learning and Epistemology Group. 1986.

Goldenberg, E. Paul. "Iconic Programming: A Strategy and Metalanguage that Helps One Learn the *Hard Parts* of Logo." Logo '86 Proceedings. Cambridge: MIT Learning and Epistemology Group. 1986.

Papert Seymour. "Computer as Mudpie." from Intelligent Schoolhouse: Readings on

Computers and Learning, edited by Dale Peterson. Reston, Va.: reston Publishing Co. Inc., 1984.

Papert, Seymour. "Computer Criticism vs. Technocentric Thinking." from the Logo '85 Theoretical Papers. Cambridge: MIT Learning and Epistemology Group., 1985.

Papert, Seymour. "Interview with Seymour Papert." Omni Magazine, October 1985.

Papert, Seymour. "New Theories for New Learnings." School Psychology Review, Volume 18. Number 4., 1984.

Papert, Seymour. Mindstorms: Children, Computers, and Powerful Ideas. New York: Basic Books, Inc., 1980.

Papert, Seymour. Writings in The Computer in the School: Tutor, Tool, Tutee. Robert P. Taylor, Editor. New York: Teachers College Press., 1980.

Solomon, Cynthia, Brian Harvey, Margaret Minsky. Logoworks. New York: Byte/McGraw-Hill., 1986.

Thornburg, David. Beyond Turtle Graphics. Menlo Park, CA: Addison Wesley., 1986.

Turkle, Sherry. The Second Self: Computers and the Human Spirit. New York: Simon and Schuster (Touchstone)., 1984.

Watt, Daniel. Teacher-Made Microworlds: or Training Teachers to Use Logo vs. Training them to Teach It. Antrim, NH: Educational Alternatives., 1985

Watt, Daniel, Molly Watt. Teaching With Logo: Building Blocks for Learning. Menlo Park, CA: Addison Wesley., 1986.

Gary S. Stager
Director of Training
Network for Action in Microcomputer Education
12 Locust Place
Wayne, N.J. 07470
HOME: (201) 831-0133
OFFICE: (201) 694-7800

Integrating Logo into the Elementary School Curriculum

Dr. Chris Templar
Johnson Bible College
Knoxville, Tennessee

Many elementary school teachers are experiencing severe pressure to incorporate more and more subjects and additional materials into the curriculum. As a result, when the suggestion is made that Logo or some form of computer education should form a part of the elementary child's experience their desire is to throw up their hands in horror and say, "There is no time, I'm too busy already."

A second factor often faced by those who wish to make Logo a part of the curriculum involves the problem of maintaining the discovery oriented nature of Logo and yet providing enough structure so that both insecure and pressured teachers will be willing to embark upon the adventure of teaching Logo.

It was with these two factors in mind that we began to make plans to develop a curriculum for kindergarten through fifth grade teachers for the Knox County School District. Knox County surrounds the city of Knoxville in Tennessee. The school system already had in place a computer curriculum. This curriculum allowed for the teaching of Logo at each grade level, however, both the basic approach and the necessary materials still needed to be produced.

After looking at our situation it was decided to develop curriculum guides which had a three-fold approach as follows:

- a series of units to include prepared worksheets that would introduce the new primitives
- a series of activities and procedures which would directly integrate with existing curriculum areas
- the outline for a microworld which would be based around a major curriculum area, and tie in skills from other disciplines and which would allow for development in Logo skills and extended exploration.

In each of these areas the goal was to provide adequate help for the inexperienced or insecure teacher while allowing freedom to the student to discover and explore in a Logo rich environment. This workshop will show the ways in which this curriculum was developed and also some of the training tools which were assembled to help in teacher training.

Parallel with the development of curriculum materials has been a steady emphasis on teacher training and help. Some of the ways in which this has been undertaken are:

- special classes for lead teachers
- make-it-take and other in-service sessions for teachers
- classes at individual schools during school time for grade level teachers

- after school training sessions led by the lead teachers at individual schools
- coordination of the program by a district-wide computer specialist who was involved in the curriculum development

The following charts show the selected primatives for each grade level and the microworld focus. At the time this paper was written only kindergarten through third grade materials have been completed and implemented, therefore information is only supplied for these grades.

List of Primatives Introduced at each Grade Level

Grade	Primatives
Kindergarten	Instant Logo Big Trak / Compurobot FORWARD FD BACK BK LEFT LT RIGHT RT
First Grade	PRINT TEXT HIDETURTLE HT PENCOLOR SHOWTURTLE ST COLORSET PENUP PU BACKGROUND PENDOWN PD *PROCEDURES* TO ... END
Second Grade	REPEAT CLEARSCREEN SLOW WAIT HOME CLEAN PAT *VARIABLES*
Third Grade	PRINT FULLTEXT CLEARTEXT WORD CHAR (32) SENTENCE FIRST BUTFIRST LAST BUTLAST PRINT [ ];

Microworld Focus by Grade Level		
Grade	Theme	Major Emphasis
Kindergarten	Logo Alphabet	Straight line letters of the alphabet, recognition, drawing and tactile skills.
First Grade	Fun with Words	Early reading skills, emphasis on early primer words and language skills
Second Grade	Let's Build Turtle Land	Tangrams, pentominoes, shape blocks, symmetry, creative writing skills, cartesian coordinates with positive numbers, simple graphs
Third Grade	This Land is Our Land	Regions of the United States, local culture, map reading activities, signs and codes, flags, quilts, geometry of squares and triangles, poetry

Copyright C 1987 Leslie F. Thyberg

THE BASIS FOR A PROCESS APPROACH TO PROGRAMMING IN LOGO

Leslie F. Thyberg
Learning Research and Development Center
University of Pittsburgh

Introduction

Producing effective thinkers and problem solvers is a major emphasis of current educational reform. A weakness in most approaches designed to meet this demand is that problem solving and critical thinking skills are not applied to the children's personal experiences. If educators are to help children become effective thinkers, they must create an environment in which opportunities for reasoning, making choices, exchanging ideas, and reflecting are integrated into the classroom organization. Quality learning happens when a child experiences a personal interaction with knowledge. Papert, in Mindstorms, relates this type of learning to Piaget's theory of assimilation. He explains that we must connect formal knowledge with the sensorischemata of a child. (Papert, 1980)

The teaching of critical thinking and problem solving can be integrated with students' personal experience. Children are constantly presented with problem situations in the classroom. Solving a math equation, revising a piece of writing, feeling anger toward a friend, and creating original procedures using Logo are but a few examples. Integrating a problem solving approach in the classroom organization and facilitating its use in every problem situation can enable a child to personally experience the problem solving process. This empowers children with the ability to become more effective thinkers.

Mine is a process oriented classroom with a unique vision. This multi-age classroom is housed in the Learning Research and Development Center at the University of Pittsburgh. It is an experimental classroom whose roots began in the widely disseminated instructional program known as the Adaptive Learning Environments Model (ALEM). This model strongly emphasizes individualization and the empowering of the child as learner. The kindergarten through third grade classroom has sixty children and fifteen IBM PC jrs, complete with graphics printers and color monitors. LCSI Logo is used with even the youngest of our students.

My efforts to create a "model microworld" led me to reflect on the work of individuals such as Papert, Bransford and Stein (Bransford, 1984), and Graves (Graves, 1981), and the models for learning that they have put forth. One phrase in particular caught my attention during my studies:

> "Children want to write. For years we have underestimated their urge to make marks on paper. We have underestimated that urge because of a lack of understanding of the writing process..." (Graves, 1981, p. 17)

The basis of my presentation is a paraphrase of the preceding quote. I propose that "children want to program and that for years we in the Logo community have underestimated the significance of the doodles, drawings, and debuggings that they have been placing on the computer screen." I believe that what is missing is the lack of an understanding of the programming process. Awareness of, and a study of, process is one of the most fruitful approaches to the teaching and learning, not only of written

expression, but of any activity requiring active cognition.

My training and experience as a reading and language arts instructor has led me to approach the teaching of Logo as I do the teaching of writing. To borrow Grave's expression - it is a "process approach." The genesis of this approach can be found in Gordon S. Rohman's 1964 article, "Prewriting", which drew lessons from the way professional writers write. The idea was profoundly extended by Donald M. Murray in his book, A Writer Teaches Writing (Murray, 1968). This theory was first applied to school writing in Janet Emig's 1971 research report, "The Composing Processes of Twelfth Graders" (Emig, 1971). Over the years it has trickled down through the secondary grades to middle school. Graves took it a step further by teaching primary school children how to write at his Writing Process Laboratory of the University of New Hampshire. The study of the writing process seems to have begun at the top and worked its way down to the primary level (Graves, 1981).

It is my belief and experience that the notable principles of teaching writing are directly applicable to the art of teaching Logo programming. The discoveries of the advocates of the process approach have relevance to teaching and learning at all levels - entry, secondary, or tertiary. The process approach to teaching Logo is the proposed subject of this presentation.

Problem Solving and The Process Approach

There is no set order to the stages of the writing process; the process is like children themselves - highly idiosyncratic and situation dependent. Nevertheless, there are identifiable ingredients in the writing process.

These stages or ingredients are essentially the same in writing as they are in programming with Logo. In my classroom environment we have a Logo Center and a Writing Center. Both have the process steps posted in plain view for all to see. Briefly described, the ingredients to this approach are:

1) Topic Choice: Whether writing an essay or creating a Logo project, both begin with the selection of a topic. Topic ideas are provided in the Turtle Center and at the Writing Center. Topics may overlap. A topic may be embedded in a previous project. The child may choose to collaborate with a peer or select a topic from a brainstorming session or a list labeled "future topics" that he or she keeps tucked away in a Turtle Folder or notebook;

2) Rehearsal: Rehearsal refers to the conscious or unconscious preparation writers and programmers make for what is to follow. Daydreaming, sketching, working with manipulatives, playing turtle are all part of the rehearsal process;

3) Composing: Composing includes selection of information and mechanics in order to assemble parts into a whole;

4) Reading: Reading the written text is part of the writing process; in the programming process, "reading" refers either to actually seeing one's program run or stepping through the program to hunt down errors and/or to check the appropriateness of the program's content; and

5) Revision: Revision, literally "seeing again", ranges from a simple adjustment for a small bug to making a major addition, deletion, and/or reorganization of subprocedures.

The process approach takes time - lots of it - but it is a worthwhile investment. How is all this accomplished? I have found Bransford and Stein's "IDEAL Problem Solver" model to be a clear and simple approach that synthesizes many of my views on how to get students involved in cognitive processes (Bransford, 1984). It is a model that is easily integrated into any classroom organization. The five components of this model include:

1) I - Identify the problem

2) D - Define and represent the problem with precision

3) E - Explore possible strategies

4) A - Act on a strategy

5) L - Look at the effects

Note that the first letter of each component spells the word "ideal". The five steps of the "IDEAL" acronym correlate directly with the five steps of the writing process. This "IDEAL" technique has become an integral part of my classroom. All problem situations are subjected to the "IDEAL" scheme, including interpersonal and behavioral problems. Thus, when a student approaches a Logo problem, she finds it quite natural to begin with "I" and work through the steps to a satisfactory solution.

The presentation itself will highlight the following topics:

- The rationale for the integration of a problem solving model into a classroom;
- An explanation of Bransford and Stein's "IDEAL" problem solving model;
- An explanation of Grave's model for Writing as a Process;
- Methods for introducing these models to students; and
- Strategies for application of the models in the cross-curricular context of Logo programming.

Research that Graves and others have conducted indicates that when writers write what they know and care about, quantity and quality significantly improve. Children in such settings have been found to be more effective writers, with a solid grasp of the mechanics of the English language and an understanding of the applications of techniques traditionally taught in isolation (Graves, 1981, Thyberg, 1981). It seems reasonable to think that a comparable phenomena should occur by teaching Logo in the same manner. While I am a practitioner, not a researcher, I am making an effort to systematically observe my students' behaviors and outcomes. Results of these observations will be shared at the time of the presentation.

I can state that outside observers of this environment - ranging from the Minister of Education from Iceland to Seymour Papert himself (keynote address, Logo '84) have cited this as an effective Logo learning environment. I believe I have devised a direct and simple means for teaching children to control their learning situations. Empowering students to perceive themselves as learners and to be able to act on that perception is what I believe my role as educator should be.

The Carnegie Task Force on Teaching as a Profession stated that:

> "An economy based upon people who think for a living requires schools dedicated to the creation of environments in which students become very adept at thinking for themselves, places where they master the art of learning and acquire a strong taste for it." (Carnegie, 1986,p. 45)

I believe that applying to Logo programming such models as the "Writing as Process" model and the "IDEAL" problem solving model is another step towards answering this challenge.

References

John D. Bransford and B. S. Stein. *The IDEAL Problem Solver.* New York:W.H. Freemand & Co., 1984.

Carnegie Task Force. Carnegie Task Force on Teaching as a Profession, A Nation Prepared: Teacher for the 21st Century. *Chronicle of Higher Education*, 21 May 1986, 1-45.

Janet Emig. The Composing Processes of Twelfth Graders. Research Report.

Donald Graves, edited by R.D. Washe. *Donald Graves in Australia - Children Want to Write.* Australia:Primary English Teaching Association, 1981.

Donald Murray. *A Writer Teaches Writing.* Boston:Houghton Mifflin Co., 1968.

Seymour Papert. *Mindstorms.* New York:Basic Books, 1980.

Leslie F. Thyberg. A Writer's Workshop: Motivating Children To Write. Master's thesis, University of Pittsburgh, 1981.

Copyright C 1987 Leslie F. Thyberg

YOUNG USERS OF LOGO- THE CHALLENGE AND THE OPPORTUNITY

Leslie F. Thyberg
Learning Research and Development Center/Falk Laboratory School
University of Pittsburgh

Introduction

What is the challenge of teaching Logo to young students? Recent figures on computer use in the schools tell us that about three-quarters of the schools having computers leave them idle more than half the school day. Seventy-six percent of the children using computers at the elementary level get less than thirty minutes of hands-on time per week. Of those elementary students who have actual computer contact, a remarkably small number of them are primary level. Still fewer preschool age children have any computer contact at all.

It is my view that the foremost challenge, if any significant impact is to be made on young Logo users, is to provide considerably more computer access. I am not a magician and have no instant solutions to this particular dilemma. I am, however, a practitioner, with over five years of experience in working with young children and Logo. In that capacity, I routinely encounter those who question technology's effects on children. I tell them that we should be examining the quality and quantity of children's experiences with the computer. This seems especially true when evaluating the impact of Logo on young students.

Another related challenge is to overcome the QWERTY paradigm of well-intentioned teachers who insist that until students can read, and, in turn, have mastered keyboarding skills, they should not be allowed to use the computer in any fashion other than to be subjected to passive "drill and kill" type packages (space bar being the only known key). Many teachers I have met have been concerned that Logo requires "too much typing for a young child". But reading and language specialists readily recognize that children perceive themselves as writers long before they perceive themselves as readers. Why else would children scribble on the walls with such a passion?! In a similar fashion, children can be encouraged to perceive themselves as capable of actively manipulating the computer even before they perceive of themselves as expert typists.

Still other teachers present a third challenge by expressing a reluctance about having their students work with Logo, not because they believe it to be too difficult for their students but because *as teachers* they encounter difficulties with classroom management ("How can I teach a math lesson and still help my kids do Logo?") In my experience, Logo can very readily be integrated into a primary and/or preschool curriculum. Certainly computer play can and should come in right along with the mud pies. Furthermore, as a language that has easy access for young users, Logo is quite effective as a means of providing opportunities for growth in many areas of learning.

These challenges are legitimate. Issues of the quantity and quality of access, of keyboard mechanics, and of classroom integration are ones that every teacher of young Logo users must face. While there are many issues that one could address under the "challenge and opportunity" umbrella, I am choosing to explore only these few - in a general and pragmatic fashion - based on my experiences as a primary level Logo teacher. This presentation will be largely anecdotal in nature, with specific examples

and strategies I have used in implementing what is considered to be an effective, even "high-powered", Logo program for children ages four through nine.

My Environment

Our laboratory school aims to be on the cutting edge of education. At the University of Pittsburgh, the coeducational Falk Laboratory School tries to put into action the latest in educational research, teaching techniques, curriculum development, and classroom management theories for children from nursery through middle school. The school offers multi-age homerooms, cooperative teaching, and a non-graded program that allows the child to progress at his/her own rate. An emphasis is placed on character development, critical thinking skills, and creativity.

The experimental classroom which I direct has sixty children, ages four through nine, kindergarten through third grade. There is an instructional staff of five adults, including an aide and university practicum students. The overall instructional program derives from the assumption that children learn better if educational programs adapt to them as individuals, and especially as individuals who play an active role in the learning process. The program emphasizes education in the basic skills, as well as the development of self-confidence, intrinsic motivation, and the ability to plan and manage one's own learning. Students are permitted to plan their own school day. This is called self-scheduling. Beginning with small blocks of time and relatively few alternatives, students learn how to organize more and more tasks into increasingly longer time spans. Subject areas include math, reading, science, and spelling. Time also is devoted to creative play, to exploratory learning activities, and to perceptual skill development. The classroom has fifteen IBM PC jrs. with graphics printers and color monitors. LCSI Logo is used with even the youngest of our students.

How I Manage Sixty Children and Fifteen Computers

The staff in our classroom plans and evaluates an informal computer curriculum for the students. Children have the opportunity to choose the computer activities in which they would like to participate. They are encouraged to ask questions, to learn, and to discover new programming concepts and strategies on their own. The intent of the program is to help children become self-directed, capable, and purposeful computer users. The teachers strive to provide an enriched environment in which each child can grow. In such an environment the children gain confidence in making decisions and in solving problems.

There are many strategies I use to introduce children to the computer. From the beginning, the emphasis of any computer lesson is on the fact that the computer is just a machine - it does not have a brain! The children learn that they must be responsible for the computer's operation. This responsibility extends beyond that of simply telling the turtle what to do, and includes the whole gamut of issues involving the thoughtful and considerate use of computers in a community of other users.

One such mechanism for teaching this responsibility is that younger students work with a teacher or peer tutors. They are given extensive amounts of time to explore and experiment. The older children in the classroom receive group instruction through the mechanism of signing up for topical seminars in which they might be interested. Most of the older children have a "license" to turn the computer on, insert and eject disks, and to work as a "consultant" with our wide variety of software.

I have devised a licensing system that has proven effective as a management scheme and as a metaphor for working with the Logo turtle. This licensing procedure is one of the most helpful tools I have for monitoring computer use and responsibility. To encourage responsible use, all children are issued "Learner's Permits". I work with the children to help them gain an understanding of responsible computer use (e.g., not pounding or jamming the keys on the keyboard). I also work on developing their sense of directionality - left and right, and so forth. Being a "Logo learner" is analogous to learning to drive a car. While one might feel somewhat anxious or intimidated at first, he/she is the driver from the start. Having young students exercise control over the turtle can help them begin to develop that sense of "empowerment".

When the child achieves a degree of proficiency in being able to manipulate the turtle on the screen, a "Driver's Test" can be given, either orally or in written form, to examine the child's proficiency. This test can be at the teacher's or the student's initiative. I give a driving test using a target game developed by an older student. It is similar to Dan Watt's "Shoot" (Watt, 1983). Passing the "driver's test" entitles the child to load and unload his or her own disks, as well as to do the same for other "unlicensed" peers. This frees me from the task of continually loading and unloading disks. Furthermore, my students are learning responsibility for the equipment and their behavior with computers.

The highest level is the "chauffeur's license", which gives the operator "consulting" privileges as well as permission to operate other peripherals (e.g., graphics printers and utilities programs). Just as automobile licenses can be suspended for operating violations, terminal licenses can also be temporarily suspended for violations of rules of use. Incidentally, I have only suspended a license twice in five years. My favorite episode involved a six year old who was angry with his friends and wrote a charming, albeit inappropriate, procedure called "To Poop". It was an imaginative use of Logo's list processing capabilities!

Another outcome of the licensing strategy is that it has promoted social interaction among the children. The basis of the computer instruction in my classroom is exploration. Although an adult is always present for guidance and encouragement, the children are given time to discover on their own and to interact with each other. The discoveries are made more significant when the child is able to share the new information with a peer. More experienced users are given the opportunity to teach the novices and newcomers to the classroom. They take the responsibility very seriously. To enforce this constructive exchange of ideas, I invented a management rule - "Ask Three Before Me" (Thyberg, 1985). Children are required to ask three other peers for assistance, before they can ask an adult for help. Once again, I am freed up to attend to those children who truly need my assistance. Beyond helping me manage my time, "Ask Three Before Me" has proven to be a easy, but effective scheme for causing the "wildfire" phenomenon, in which one student catches on to a new idea or discovers the cause of some bedevilling bug - and can't wait to pass it along to her peers.

While some teachers have found this to be an issue, I personally have had little difficulty in teaching my students how to use the keyboard. I have used a number of different strategies. For example, one is to label the most obvious keys (F, D, B, K, etc.) with little press apply dots so the children are better able to locate them. Another is to use a variation of Instant following the guidelines set forth by Bull and Cochran in their Logo '86 address, "The Issue of Instant Logo". They state:

> An effective Instant Logo program should have the following characteristics: capability for entry of any number; appropriate turtle step length; appropriate turtle step speed; visible results with small numbers; messages which tell what is happening on the screen; meaningful error messages; easy exit and resistance to crashing (Cochran, 1986, p. 95).

The computer has been integrated into all facets of the school curriculum. On occasion, Logo is used to support and extend lessons that are completed in the classroom. More detailed discussions of two curricular areas, math and language arts, will be covered at the time of the presentation.

Briefly, the language arts curriculum has been greatly enhanced by the use of computers. For example, as a part of reading readiness children learn some of the basic principles of word processing to complement the graphics which they generate with Logo. Story telling and writing activities are effective language learning experiences for preschool and primary age children. Logo graphics and inventing adventures for the turtle are wonderful motivators around which to spin stories. An additional bonus is that even kindergarteners quickly learn how to recognize words and error messages. The older students serve as editors and assist in the weekly publication of the "Turtle Times".

As a result, students are discovering numerous applications for Logo. Group projects involving Logo have frequently been included in our weekly plans. The students have produced ABC books and quilts as group projects. In addition to the newspaper, birthday and get well cards have also been designed. Musical recitals have been performed. The children take great pride in these productions! They serve as tremendous *"PR"* with our parents as well.

Our aim is to give the children a positive introduction to a machine they will have to be familiar with as they grow up. With knowledge of the proper terminology and handling of the equipment, the children develop confidence and interest in dealing with the computer. They will be able to grow up with a healthy attitude toward computers. It is our hope that they will understand that the computer can work for them, a tool to help them in everyday tasks. My students may not be consciously aware of this goal; for the time being they see the computer as something that is fun to work and play with. But this is a good place to begin.

References

Paula Cochran and Glen Bull. The Issue of Instant Logo . Paper presented at the Logo '86 meeting of the
1986.

Leslie F. Thyberg. Too Many Questions, Not Enough Me. *National Logo Exchange*, May 1985, 1-3.

Daniel Watt. *Learning With Logo.* New York:Byte Books / McGraw-Hill, 1983.

Logo Beyond the Square

Steve Tipps
West Professor of Education
Midwestern State University, Wichita Falls, Texas
Columnist, The Logo Exchange

When Logo was developed at MIT, the team of scholars envisioned a language which would bring the power of computers to many people--not just scientists and engineers with strong mathematics backgrounds. Logo, the product of a decade-long collaboration of computer scientists, mathematicians, and educators, is a "microworld" (Papert, 1980) in which learners explore exciting and powerful ideas. While grappling with the task of drawing a square with an electronic object called a "turtle," children or adults learn about the properties of a square.

It has four sides.
Sides are be any length, but all must be equal in length.
It has four angles.
All the angles are equal, and all must be 90 degrees.

This knowledge is gained in an inductive, experiential way. Rather than being taught the square facts for recall, the square facts are internalized actions and stimulate drawing of other shapes and understanding of geometric concepts.

Since 1981, the debate about why and how Logo might be implemented in schools has been loud and lengthy. What is all the fuss about Logo? Some people see the turtle draw a square and wonder why they should bother with Logo. A pencil on paper can produce a square at must less cost. Many graphics packages produce shapes just as square. Some of the software doesn't even require any programming and is "easier" to use than Logo. Other people see the value of learning by doing and are initially excited about the possibilities. Several weeks and thousands of squares later, they cannot think what there is to learn with Logo beyond the square.

To answer questions about Logo and the utility of drawing a square, the objective and outcome of drawing a square with Logo must be examined. In answering the question of "what next?" a broader perspective on Logo is required.

Another Square Look

Making a square is a "project" or activity in Logo. The learning is not the square itself. Learning comes about in two forms: in the process leading to the square and in the understanding after drawing the square. The process of drawing the square is one of description at several levels of representation.

Body / concrete level:
Walk forward, turn, walk forward, turn, walk forward, turn, walk forward, turn.
Object / concrete level:
Build a road with four blocks the same size.
Move a toy truck on the road by rolling and turning four times.

Symbolic / abstract level 1:
Make the turtle move by typing commands four times.

```
FORWARD 50
RIGHT 90
```

Abstract level 2:
Find the pattern in the turtle moves and use repetition.

```
REPEAT 4 [ FORWARD 50 RIGHT 90]
```

Abstract level 3:
Create a new command which stands for all the movements.

```
TO SQUARE
REPEAT 4 [ FORWARD 50 RIGHT 90]
END
```

Draw the square by typing the command SQUARE.

Although learners may begin with the task at the psychomotor level, they learn through experience to describe the project at more abstract levels. Logo, through its structure, encourages learners to make their descriptions more parsimonious. SQUARE subsumes all the knowledge which has been gained. Words are labels for experience.

SQUARE also is the culmination of an experimental process which included many hypotheses and many trials. Exactly what angle is needed? Will 95 do as well as 90 for the angle? What about 91? Could I make a smaller square and a larger one? Most of the trials involve a good number of errors. Mistakes are expected and desired in Logo. By making mistakes and correcting them, the learner constructs both the square and the understanding of a square.

After making many squares, the geometric properties of a square are "deep," or operational knowledge. Rather than a lesson on right angles, the learner only has to attach the label "right angle" to experience which has been explored fully. Papert (1980) hopes that learning mathematics and geometry in this way will be as natural as a child learning a language.

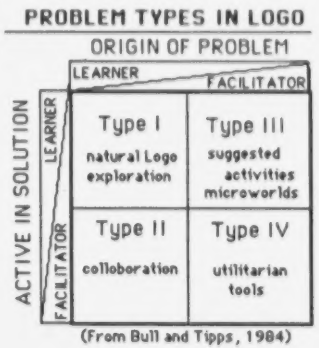
Pencils and graphics packages also make squares, but the objective is not the same. The educational goal of Logo was never to draw a square but to know more about it. The square is, in many ways, trivial. Describing problems, making mistakes and correcting them, constructing hypotheses and testing them, and finally, drawing conclusions about the properties and extending these ideas are not trivial. The square is the visible evidence of learning which may not be visible in other ways for several years. Each new Logo project makes its own demands on thinking and offers different possibilities for extensions. The learning cycle only begins with the square. Can I also make a rectangle" What is different and what is the same? How is a triangle like a hexagon? What angle is needed to draw a circle? How do I make a larger or smaller circle? These questions come naturally to many children. Others need questions or probes to help them focus on bigger ideas. Knowing when and how to intervene to bring deep knowledge to the surface is a residing question in Logodom.

Four Square Problems in Logo

Making a square is often one of the first projects attempted in Logo. After an hour in a room of first time Logo users, whether they are 7 or 70, screens are fully of squares and square projects. Something in the collective unconscious must motivate creation of squares. Even though adults already know how to draw squares with pencils and know, or once knew, the size of right angles, they seem impelled to make squares and reconstruct or rediscover squareness.

Squares are prime examples of what have been called Type I Logo problems (Bull and Tipps, 1984). Type I problems are projects which learners invent and solve for themselves. Type I problems provide the model for introducing Logo to learners. They are non-directed, but not non-directional. The facilitator may be around to help with a few casual nudges, clues, and hints, but the learner is the creator. A desperately needed command, such as REPEAT or SENTENCE, or a challenging question may help overcome a difficulty or consolidate learning.

If Type I problems exist, what other problem types also are available? A four-square grid shows four types of problems which can be broached with Logo. The problem can originate with the Learner or with the Facilitator of Learning. The problem can be solved by the Learner or by the Facilitator. In practice, responsibility for the problem and the solution are often shared by both participants. However, the learner or the facilitator may be more active at different times and in different situations. Each of the four squares in the chart describes a different type of interaction which is possible in a Logo environment.



(From Bull and Tipps, 1984)

Problem Type I, called "natural" Logo, is learner originated and learner solved. The facilitator is a support in this kind of project. Exploration of ideas, mistakes, and problem-solving are the core of Type I Problems. Each new accomplishment brings about another challenge. Intervention is not constant, although sharing of ideas is.

Type II Problems are also learner originated, but the facilitator is must more involved as a resource and consultant. As students move into bigger projects, they find themselves in situations without the skills to complete a project. The facilitator assists by providing coding to match the learner's description of the problem. Sometimes, the facilitator helps the learner think through the problem. The thinking behind the solutions should still be the learner's as much as possible, but the facilitator steps in to provide technical help with Logo.

Type III projects are developed by the facilitator for solution by the learner. Several Type III Problems are possible, the most common being Logo activities which teachers organize for students. These may be worksheets, task cards, or booklets. Type III projects may be necessary because learners have a hard time inventing new problems for themselves and because one-on-one help is not always feasible. Another kind of Type III Problems are invented microworlds in which the facilitator creates procedures to explore a particular idea. Natural Logo has commands such as FORWARD, BACK, RIGHT, and LEFT for exploring distance and direction. But a facilitator who wants students to explore concepts of area, or sentence types, or maps, or probability, or adjectives, or populations, or electrical fields, or graphing, or types of music, or any other topic can exend Logo and enable students to "play" with the concepts in a Logo environment.

Type IV Problems are facilitator originated and facilitator solved. In many ways they are like Type I problems; the difference is in utility. Type IV problems are of a tool-making nature. The facilitator may need new commands to demonstrate graphing or to pick student helpers randomly. The motive is utilitarian rather than exploratory. Some times learners slip over into Type IV projects themselves when they have to make tools for a project.

Not discussed fully in the original presentation of the four problem types (Bull and Tipps, 1984) is the goal of Type II, Type III, and Type IV Problems which is to stimulate new Type I projects. The success of intervening and assisting with commands or consultation (Type II) is measured by the incorporation of those commands and ideas in further exploration. The purpose of task cards, assignments, and worksheets (Type III) is to inspire Type I projects. Type III suggestions are starters that should help students resume Type I problem-solving. Good microworlds (also Type III) are not limited, rigid procedures aimed at finding answers; they are open-ended explorations in which new problems, never imagined by the facilitator, are possible. These microworlds, if they are well conceived, also lead children into Type I Problems in which they invent new activities for the commands.

Too Square

The four problem types were suggested to describe a variety of appropriate ways Logo could be implemented. Teachers often felt they were not living up to Papert's dictum of "learning without teaching" if they did anything but allow free exploration. Students can learn much with direct instruction, but they probably do not learn much without having a direction. Many teachers and students got stuck in Type I problems and felt they had exhausted Logo. When students in the fifth grade moan "Oh, no, not Logo again. I already know how to draw a square," they have a valid point. Too many squares have been drawn without a sense of purpose or a larger perspective. The square has limited our vision.

In the last three years, the focus on Logo has increasingly changed from programming to problem-solving. The evolution is apparent in articles in The Logo Exchange which feature greater content orientation and more tool making. The same is also found in the "Logo Corner" of The Computing Teacher. Logo practice in the classroom has not kept pace with this change in the literature. Every year, teachers and students seem to start again by drawing squares. Many commercial materials are fill-in-the-blank and recipe activities--bad examples of Type III projects--which lead not to more exploration but to the next page in the workbook. While students, teachers, and schools need time to grow into more mature uses of Logo (Tipps and Bull, 1985), the time is slipping away without that maturation.

The message of Logo is not that the turtle can draw a square. The message of Logo is that teaching the turtle to draw a square is one first step toward using the computer as a powerful tool. Logo is a tool, a superb tool, in the hands of a facilitator who is clear about intent, content, and context. The intent of Logo is thinking, the content is limited only by imagination, and the context is a lively environment of shared learning and creativity.

Bibliography

Bull, Glen, & Tipps, Steve. (1984). Problem spaces in a project-oriented Logo environment. The Computing Teacher, 11 (5), 54-57.

Papert, Seymour. Mindstorms. New York: Basic Books, 1980.

Tipps, Steve, & Bull, Glen. (1985). Teachers need time for turtles: planning for teacher development with Logo. Logo in the Schools (Cleborne Maddux, Ed.). New York: Haworth Press, 273-281.

Logo-Like Music Environments

Rena B. Upitis
Faculty of Education
Queen's University
Kingston, Ontario
Canada K7L 3N6

Countless powerful computer tools for music education have been developed. Of these, several have been designed with the purpose of actively engaging the student in such musical explorations as improvisation, composition, and the analysis of musical structures. Logo Music tools fall into this category. Many educators would agree that the development of such musical tools holds great promise for public school music education.

Even with the emergence of promising tools like Logo Music and other similar programming languages or software packages, teachers and students are unlikely to make full use of this type of software without a supporting musical environment. There are at least two reasons for this. First, most children spend little time playing with musical ideas and creating original music, particularly once they begin formal schooling. (Note that this is in sharp contrast to a child who first comes in contact with a computer graphics language, since virtually all children have tried their hand at drawing or painting; likewise for word processing programs - children experiment with spoken and written language from a very early age.) Thus, it is argued here that teachers will first need to provide a "musical playground" where children can improvise and compose without a computer. Then, when children approach composition in a procedural fashion at the computer, they will already have some ideas about how musical segments can be transformed and manipulated, based on their earlier playful explorations. The second reason for learning how to create a dynamic musical environment is that the computer alone cannot provide a full musical experience even if children have first learned about music by exploring musical ideas away from the computer. The computer cannot replace what children learn and produce with their own bodies, nor can it provide the kind of learning which is involved in playing a classical instrument. Rather, computer-based learning

along with other open-ended music settings serve to enhance one another, and together contribute towards building a living musical environment. This type of musical environment is, in fact, a "Logo environment" in the fullest sense: a place where a child can direct his or her own learning in a meaningful way that leads to personal insights into the very heart of the subject under exploration. The remainder of this paper will describe what such a Logo-like musical environment or playground might look like.

The music environment described in the present paper is based on a music setting in a high-density computer school in Roxbury, Massachusetts. A team of researchers from the Massachusetts Institute of Technology collaborate with teachers and students at the school on an ongoing basis in designing new learning environments. One such environment which was developed in this way during the 1985-86 school year is perhaps best characterized as a musical playground, where children are able to make music to communicate messages, stories, ideas and feelings using a variety of musical media.

While it is true that computers played a significant role in the atmosphere of the musical playground it is also true that the computer presence cannot solely account for the excitement and learning which transpired in this musical environment. Arguably the most important aspect of the music setting was that children created their own music. Children improvised, composed, notated, listened, and performed, and the computer provided one strong tool for achieving these goals. Further, by using the computer as one of a number of materials for making and understanding music, the computer came to be regarded not as something special or different, but rather, as a useful tool for some purposes but not others. The latter point is illustrated by one child's comment: "Last night I thought of a tune I want to try. I think I'll do it on the computer....or maybe I should use bells first." The child was in a position to choose what she viewed as the most useful tool for the particular task (a task which she had set for herself to explore).

Having said that the computer should not be the only important feature of a musical playground, it is nevertheless a valuable one. One of the times that the computer is an appropriate tool is when children have a piece of music in mind which is too complicated for them to play and manipulate on a conventional instrument. By using input devices such as the Koala Pad or Mouse or even a synthesizer keyboard, children can create complete musical patterns with relative ease. The computer is also a valuable tool for making musical manipulations of the original idea. For instance, a number of music composition programs enable children to compose musical segments or motifs and then manipulate those

segments using musical transformations such as inversion, retrograde, transposition and augmentation. The computer can also be used in the music classroom as a new performing instrument, where children become involved in the process of inventing new sounds. The computer as a music-making device can also be combined with other technologies (e.g., computer animation and text) to create new forms of expression.

Movement improvisation should be a major part of the children's musical play experiences. Many of the movement activities at the school were based on the principles of Emile Jaques-Dalcroze, a Swiss musician who developed many techniques for teaching musical concepts through gross and fine motor movements. For instance, one improvisatory movement exercise, which was used often during the school year as different musical concepts were explored, was based on children's names. At first, children were simply asked to "Show your name with your head, saying your name as you move your head." A variety of head movements were offered by the children, including rotating the head for a name like Melanie, or a quick left to right movement for a two syllable name like Edward. This activity was enormously successful for several reasons. First, every child knows his or her name, including the musical content of the name, such as the rhythm pattern, stress, and tempo at which the name is "played." Children are also well aware of the way in which they can move their heads. Thus, the combination of two things they know well makes it possible for them to generate a great variety of ways of their own to "play their names." Further, more complex musical notions such as polyphony and harmony were explored using names, by having each of three or four groups of children sing or play a name or series of names.

Both movement improvisation and other improvisational activities were introduced to children within a structure which in some way limited the musical universe. In the case of movement, specific parts of the body were suggested. In the case of improvisation with pitched instruments, at the beginning the number of pitches were limited. Children were encouraged to pick three or four tones, thus creating their own scale, and then to manipulate those tones in every way they could imagine. Thus, children were able to make a variety of musical manipulations without the added complication of juggling many pitches harmonically. The notion of playing with only a few elements and building complex structures from simple beginnings is also one which was well-suited to using a procedural language like Logo Music.

Playing with music notation systems was also featured in the music environment. Children were encouraged to develop their own notations to

"remember your song so you can play it again next week." Their invented notations typically ranged from sticks and dashes, to letters and numbers, to pictures depicting the way in which a given instrument was to be played. The benefits of using children's spontaneous notations as a way of later introducing standard notation were considerable. By having children compose music, they developed a need for notation in order to remember their compositions or share them with others. In other words, a need to develop and standardize notational systems was generated. Common music notation took on new meaning for the children as they began to recognize its universality.

There was also a strong performance component in the program. Children performed their own compositions as well as those of others. They performed on traditional instruments (such as bells, recorder, piano and flute), on the computer, and on instruments they made from simple materials such as balloons and shower hose. Sometimes performance involved a combination of computer and non-computer instruments. Performances took place both during class time and in informal concert situations. The most popular informal concert setting was a weekly recess concert. In addition, children sang most days, often with instrumental accompaniments that they themselves improvised.

Another predominant feature of the music environment involved helping children make connections with music to other disciplines. On a rather simple level, the clear relationships between divisions and groupings of beats and meter with fractions in mathematics were explored. Even more importantly, musical correlates with such mathematical concepts as closure, repetition, and symmetry were discovered by the children. In the language domain, poetry and storytelling were linked to musical expression in the form of song. Children wrote their own ballads to describe novels they read and periods of history they studied. Similarly, often natural connections were made between music and drama. For example, many children played with puppets as a way of storytelling or role playing, often adding a musical component to their invented dramatizations.

In short, by engaging themselves in the type of musical playground described, children can come to see music as a rich and legitimate way of thinking about and interacting with their environment. The computer can play a uniquely significant role in children's musical development, but only in conjunction with additional stimulating ideas and settings to enhance the role of the computer. In this type of music-making environment, children make music their own. By having children make their own music based on material from their physical, emotional and intellectual worlds, one can begin to see how music could become a life force in the lives of children during school and beyond.

SEELOGO: A NEW PERSPECTIVE ON AN OLD FRIEND

MADGE ALEXANDER VINING

Computer Education Instructor
Eastern Connecticut State University
Willimantic, CT

Elementary teacher
J. M. Moriarty School - Norwich, CT

SeeLOGO first caught my attention at the LOGO '86 convention when a wistful turtle puppet in the exhibit hall poked its head out of its shell and looked at me. As a puppeteer, I couldn't resist chatting with "Cecil" for a few minutes, then turned my attention to its sponsor, SeeLOGO, in the K-12 Micromedia Booth to politely inquire about this product. I wasn't sure I really wanted to know. My newly-acquired knowledge of Logowriter and AppleLOGO II, fractals and perspective had facts and figures swimming in my head and I didn't feel I could absorb another new program. When the young lady told me it was an unpublished version of LOGO and they were looking for educators to preview it, I decided to take a copy home and give it a Saturday morning on my own time. (I also bought the puppet.)

I COULD NOT PUT SEELOGO DOWN!

I kept thinking, "Why hasn't someone come up with this before?" or "What a neat idea!" SeeLOGO has a fascinating collection of commands and primitives that make its use easier to learn, more fun to execute and gives a totally new outlook on building block procedures, ever changing, ever changeable, and ever intriguing!

Before describing SeeLOGO's new and exciting ideas let me give you a little background on its conception. Dan Novak and David Rosenthal are assistant professors in the department of Mathematics and Computer Science at Ithaca College, where they created and currently run a computer camp for children in the summer.

Dan is involved in working with children in problems related to mathematics and has used LOGO as one of the many tools to help children explore and learn.

David is primarily involved in designing software for use by children and teachers.

They both felt that the learning of LOGO should be simplified for youngsters so that more time could be spent using it in a learning situation - so they set out to rewrite LOGO in a version easier to teach, faster to learn, and affordable for home purchase. Now that SeeLOGO is a reality, Dan and David are working on a new version in 128K which will further enhance its capabilities.

SeeLOGO is similar to both Apple and Terrapin LOGO in that commands from either language are compatible in its implementation. Right away you can either READ or LOAD programs. CLEAR, CS, or DRAW clears the screen. SAVE and CATALOG work the same as well as PO and POTS. GOODBYE and ERALL clean the working memory, but SeeLOGO won't obey either of these commands until it asks if you REALLY want to do this and you have reaffirmed that you do. SeeLOGO eliminates all quotation marks and many of the brackets necessary in other languages but the best innovation, I feel, is being able to name a procedure after the fact. You go along drawing with the graphics (and possibly using subprocedures). If you add a move you feel is incorrect just use the UNDO command and the graphic is redrawn without the last command. Don't like two or more moves? Just UNDO 2 or UNDO 3 or more! It uncreates! When you arrive at a procedure that strikes your fancy, stop and give it a NAME!

The first noticeable difference in SeeLOGO is the new shape of the turtle. Being larger than the old triangular turtle, and having an outline form, there is no doubt which way this turtle is headed. Another nice feature is being able to change its shape through a command called CREATE and animate it through a command called STAMP. I have my own turtle shape that I have named Cecil Ogo. Cecil always faces the same direction, although the triangular turtle underneath him moves and turns, unseen, on command.

Often a syntax error gets typed into a LOGO program. If SeeLOGO balks at some command, type in "WHY?" and it politely tells where the mistake lies - whether you have used an unknown word: "I don't know that word." (with the unknown or misspelled word highlighted) or perhaps "I found an extra [". This lets the user know right away where the problem lies, eliminating trial and error search for mistakes. There are more than 30 error messages to help detect bugs.

MIRROR is a fun command. Once a procedure is named, it can be mirrored to create exciting effects - then renamed to include the mirror image. Turning the turtle 180 degrees and repeating the new command can give a kaliedoscope effect - then that procedure can be named. Can you see the endless possibilities emerging? Just wait - there's more!

Circles and arcs are built-ins. CIRCLE (R or L) takes one input: the radius of the circle. ARC (R or L) takes two inputs: first the circle radius then the number of degrees desired for the arc. No longer are arcs limited to a quarter of a circle. The turtle is always hidden while making circles and arcs, but another nice feature is the ability to hide (or show) the turtle during execution of any procedure by typing **CONTROL-Q**.

DOUBLE or **HALF** takes any named procedure and makes it larger or smaller; likewise **TRIPLE** and **THIRD**. And if you don't like these dimensions, you can choose your own size for any named procedure by using **SIZE __% PROCEDURENAME**. Great for teaching percentages! While we're at it, the command **GROW** enlarges a procedure by 50% and **GROWBY __%** lets you control the amount by which your procedure grows.

SPIN commands are exciting - they make polynomials with closure by using two inputs: the number of angles desired and the side length. **SPINR 3 50** would create a triangle (going to the right) with sides of 50 steps. Geometry becomes fun to learn!

BG and **PC** set the colors on a colored monitor and can change the colors on a black and white monitor (**SET** is eliminated.) **BG 0** makes the background black, **BG 1** changes it to white and **BG 2** reverses what is already there. This can create some startling effects using a repeat pattern. With colored monitors you can also add the colors green, violet, orange and blue as well as additional black and white by using numerals up to 10.

Ordinarily, **SeeLOGO** will not allow you to load procedures into the working memory while there are already programs there. However, you can use the **MERGE** command to do this, being careful not to duplicate names. File procedures can be changed by saving a new version under the old name. Here again, **SeeLOGO** asks if you want to erase the old version before executing your command.

The **EDIT** mode allows you to see your program drawn graphically line by line while you edit, to zero in on the bug you are looking for. However, you are free to also **BLINDEDIT** if you so choose.

MOVETO conveniently replaces setting X and Y coordinates and **TURNTO** turns the turtle toward a named point. Using **GRID** overlays a cartesian coordinate grid (with ten-step intervals) for your convenience in locating points. Can you see how this will help a frustrated geometry or algebra teacher? Typing **GRID** again erases it from the screen.

No need to use a special program to make copies of your text. **PRINTER PROCEDURENAME** does it for you with an attached printer. **PRINTER ALL** prints everything in the working memory.

A most exciting and educationally valuable feature is **SeeLOGO**'s ability to combine text and graphics anywhere and, in turn, produce a story or a multi-page article with pictures. First place the turtle wherever you wish the text to appear and type **WR [...]**. (Text is always written horizontally.) Continue by placing a picture in another location. When everything needed for a page is done, **NAME** it **PAGE1**. Continue with subsequent pages and save the entire story under a storyname. To recall a story, just **LOAD** or **READ STORYNAME**, and run the program. Striking any key will turn the pages in order.

There are too many new commands to be covered here, including game commands, algebra forms and variables. The users will discover these delightful tools on their own.

The new 128K version of **SeeLOGO**, in the works now, has a great new feature, **HELP**, which lists all of **SeeLOGO**'s commands and primitives. **HELP COMMANDNAME** zeroes in on a specific command with explanation or instructions for using it. Some of the other great ideas will include:

NOTE - takes 2 inputs for musical tone and duration. Students can become musical composers, having over 100 notes to choose from.

FEDIT - allows the capabilities of fullscreen editing.

ZAP - works like UNDO without redrawing the picture. It erases one command or one line of commands at a time.

FILL - will fill in spaces with color.

A method of hiding the turtle within a program is also being developed and the additional memory will greatly expand the capabilities of SeeLOGO programs.

SeeLOGO should not be seen as a replacement for LOGO. It does not have recursion or full list handling capabilities, but what it does, it does well: provides an environment where children (or LOGO learners of any age) can use pictures and numbers in a creative way without some of the unnecessary obstacles beginners have to go through when starting LOGO. Our thanks go out to Dan and David for their innovative achievement.

The Logo and Computers in Education Workshop:

A report on a teacher training project at
The People's Education Press, Beijing, China
by

Molly Watt and Daniel H. Watt
Gregg Lake Road,
Antrim, New Hampshire 03440

During June of 1986 at the invitation of Prof. Wu Yongxing, Director of The Curriculum and Teaching Materials Research Institute, we lead a team of seven educators and computer specialists in conducting a month long training workshop for key educators and curriculum developers in China. Our goal was an ambitious one. Inspired by our own experiences founding the Teacher Center Brookline in Brookline, Massachusetts in 1972 and the Computer Resource Center at Technical Education Research Center (TERC) in Cambridge, Massachusetts in 1980, we wanted to work with educators in China to establish an ongoing center of educational computing resources. Our dream included a continuing schedule of workshops, curriculum development and exchange of ideas, training materials and software especially suited to the educational purposes of teachers in China.

The Logo and Computers in Education Workshop was held at the People's Education Press in Beijing, China during the month of June, 1986. Zhang Wei and Steven Rasmussen demonstrate metaphors for using the Macintosh computer.

At the time of our work in Beijing, Logo and BASIC programming in English were taught in Key Schools throughout China. These schools are educating selected students to participate internationally -- which means English and computer science are essential elements of the curriculum. For three years the national student programming contests included Logo competitions. However, most schools in China do not yet have computers. We do not know what place there is at the present time for Logo and computers in education for school students in all schools in China. Perhaps the most important use of Logo at this time in China will be in the area of teacher education. By the completion of the workshop our students were as interested in our teaching methods as they were initially about the hardware we were teaching them how to use. For many teachers this was their first experience in developmental learning.

Team members included: Arthur Powell,Jr.; Dr. Donald Rasmussen; Steven Rasmussen; Flora Russ; Dr. Daniel Watt; Molly Watt; Zhang Wei

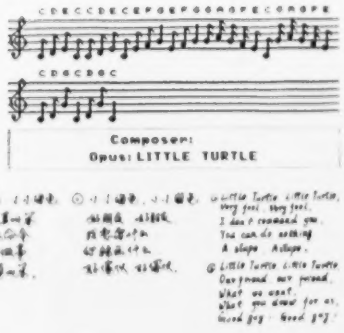
I was told a famous saying, "The teachers' vocation is the highest vocation under the sun." I think a good teacher is the highest person under the sun. What makes a good teacher? Though I couldn't answer this question perfectly, I have seen a lot of good teachers with my own eyes and heard the deeds of many good teachers with my own ears. This has made a deep impression on me.

I have never forgotten an incident that took place on my first day at primary school in Mrs. Li's class. During the course of the lesson, Mrs. Li noticed that one of the pupils looked sick and feverish. She walked to the pupil's desk and put her hand on the child's forehead. While her back was turned, another pupil in the class got up and made faces in order to make the other pupils laugh. Mrs. Li turned rapidly and glared at the pupil. "Don't do that again," she said, quietly but firmly. Then she went back to her desk and went on with the lesson. After a while, she put down her book, looked at us in silence for several minutes, then said, "My friends, we have to spend this year together and we must try to spend it together happily. You should study hard and be good pupils. I don't want you simply to promise me with words that you will be good, but I want you to show me with your hearts that here we are all part of one big family. I want to be proud of you."

At this moment, the bell rang announcing the end of the class. The boy who made faces went up to Mrs. Li and in a trembling voice said, "I'm sorry Mrs. Li." Mrs. Li patted him affectionately on the head and said, "Don't think any more about it, my son. Here we are all good friends."

Thirty years have passed. I've forgotten many things but I always remember what she said to us on my first day of primary school. It gives me the courage to work hard and never be even a bit lazy.

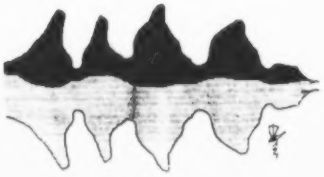
Today in our workshop, I am learning Logo and the uses of computers from the American teachers. Dan always has a good sense of humor whenever he gives us a speech. Molly and Daniel often give us vivid descriptions during the course of the lesson. On the first day of the workshop, when I saw Flora preparing all kinds of softwares and manuals herself, I thought she is just like our Chinese teachers. Maceo and Steve are always patient when they teach us. I don't define what makes a good teacher. But I think they're all good teachers. Could I say a good teacher is just like a candle which burns itself and gives light to all students?

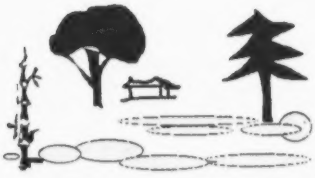
58.

Teachers used their time in the workshop to explore what good teaching is and to express their own ideas in words and music.

1. Developing a philosophy and structures to guide the use of computers with children
2. Teaching structured Logo
3. Teaching (by example) how to teach Logo
4. Demonstrating programs in a variety of educational fields
5. Leaving students with the confidence to use computers in their own teaching
6. Developing (through discussion and demonstration) criteria to evaluate software programs for use in Chinese Schools
7. Promoting the "creative use" of programs with children
8. Introducing productivity tools and their classroom applications
9. Introducing desk-top publishing

We organized the forty plus participants into teams of four or five to one Apple IIe computer. Each team was responsible for completing each task as a collaborative working experience. Each team was lead by a "student" leader. Each day Molly held a team leaders meeting to coach the leaders in coaching their teams! Our teaching methods ran the gamit in a total immersion learning experience which included leecture/demonstration, teacher challenges, conversations and on-the-spot coaching sessions, diagnostic teaching, discussion groups, collaborative programming, project sharing, journal writing and revising all work to create models to publish. (See some examples from our final 65-page project in this article.)

This workshop team is revising their Logo project to make it more readable. They use pinyin, the phonetic alphabetized written Chinese used to teach reading to young children in China.



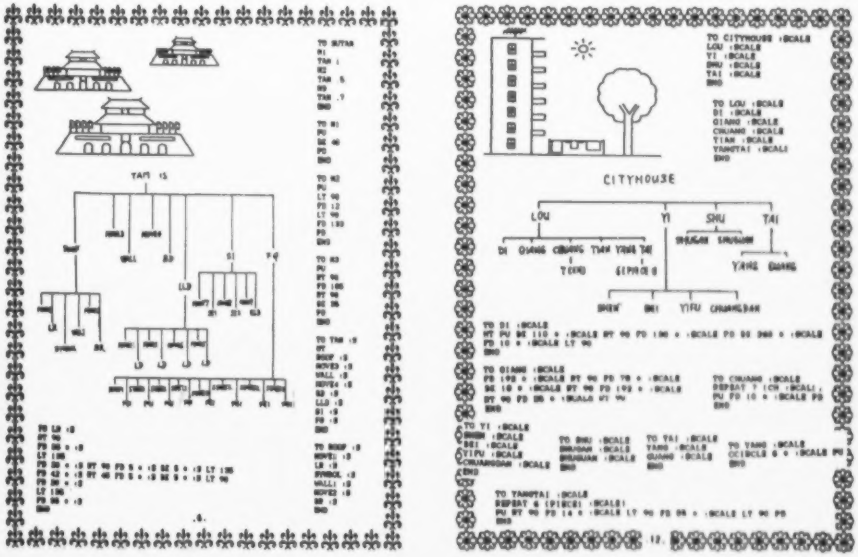


Each place in China has a particular name as does each procedure in Logo.

Our host organization, The People's Education Press, which has published 20 billion books during the past thirty years, promised to select forty key educators to participate fulltime in the workshop (8:30 A.M. - 4:00 P.M., Monday-Saturday for four weeks). In addition to providing technical support to the workshop and graciously hosting our team, they agreed to provide a place and a staff for such a center to continue after the initial workshop ended. We promised to travel at our own expense and deliver a lab of microcomputers and many model software programs and curriculum applications. Thanks to Apple Computer Company and more than fifty educational publishers, our team was able to deliver on our promise to provide the first equipment for the center.

In preparation for the workshop The People's Education Press translated Dan's book, Learning with Apple Logo plus a dozen articles on Logo for our teacher "students" to use as resources.

Our educational purposes for the workshop were complex. Our team's organizer, Dr. Donald Rassmussen summarized them in the following list:



These Beijing scenes, one of Tian An Men (Gate of Heavenly Peace) and the other of an apartment house near the workshop, are created by the teachers as models of structured programming in Logo to demonstrate to their students.

Step-by-Step

East Coast Logo Conference '87 Kickoff Session

by Daniel H. Watt and Molly Watt
Educational Alternatives
Gregg Lake Road
Antrim New Hampshire 03440
(603) 588-6734

You can't make a turtle come out,
You can't make a turtle come out,
You can call it or coax it
Or shake it or shout,
But you can't make a turtle come out, come out,
You can't make a turtle come out.

(Words and music by Malvina Reynolds, Copyright 1963)

Malvina Reynolds' delightful children's song was written long before Logo was ever invented, to help children learn to behave more sensitively toward animals in general, and pet turtles in particular. But the song has a deeper meaning as well: some things in life just can't be forced, can't be made to happen as quickly as we'd like. Some things require patience, as well as the confidence that *not* forcing them to happen, *not* trying to make them happen but rather *letting* them happen is exactly what is required.

We think that the Logo movement -- the loose collection of groups and individuals involved in promoting the use of Logo -- has now reached the level of maturity implied by Malvina Reynold's song. Most of us no longer believe that it is essential that all children in all schools must be required to learn Logo immediately. Instead, we see the wisdom of making slower progress (as a turtle does), nurturing what is already in place, with better versions of Logo, more appropriate curriculum connections and materials, more on-going training and support for teachers using Logo, and better data about what students can and are learning with Logo in classrooms. At the same time, we welcome the development by many participants here of new applications, new activities, new learning domains to be enhanced with Logo, and take a longer view of what it will take to get these new activities more widely used.

At the '86 Logo Institute Conference in Keene, New Hampshire, Tom Lough and Brian Silverman helped put things in perspective for us. Speaking as a physicist, Tom showed us how small changes introduced into a system at the right place and time, can lead to surprising and sometimes unexpected results! Brian, responding to the urge to take Logo learning "beyond turtle graphics," suggested instead, that a more appropriate rallying cry might be for a giant step "beside turtle graphics," that is, new versions of Logo, and new ways of using

Logo to create learning environments that embody the same kinds of holding power and learning potential as Logo's famous turtle.

A brief review of the program for this conference shows that many in the Logo community have been doing just that. We see evidence that many small steps sideways are being taken: new versions of Logo such as Object Logo, Music Logo, LogoWriter and Lego-Logo creating new learning possibilities beside turtle graphics; connections being made between Logo and other devices, such as videodisks, robots of various sorts, and speech and music synthesizers; and above all, applications of Logo to many aspects of the curriculum, for many grade levels, and many types of learners. What is most striking about all this is the diversity of small steps being taken, the absence of hype, and the sense of an atmosphere of confidence in which new ideas can be created and nourished, slowly and appropropriately.

Assessing Critical Aspects of Logo Learning: A small step involving collaborative classroom research

With these ideas as background, we'd like to tell you about a small research effort that we are engaged in right now. We are working with a group of eight classroom teachers and four computer coordinators in the Boston area, in an exploratory study, funded by the National Science Foundation through the Center for Learning Technology at Education Development Center, to develop ways of observing, assessing and enhancing Logo learning in upper elementary classrooms.

Our motivation to do this work comes from several sources:

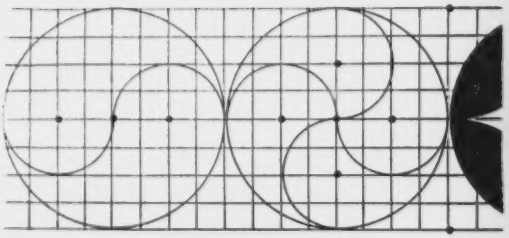
-- our close observations of our own students learning to use Logo as a programming language and an educational environment and of our own teaching interventions with them
-- requests from teachers for methods to assess their students work
-- demands from school adminstrators for evidence that Logo is being used effectively to achieve the educational goals for which it is most commonly introduced: increased student abilities in problem solving, introductory computer science, and mathematical thinking
-- disappointment with published reports that tell us how particular groups of Logo students fared on batteries of tests, but do not tell us what the same students actually learned in their classrooms
-- our own observations in classrooms around the country, that tell us that many Logo-using teachers have inadequate understanding of the powerful learning possibilities that Logo was designed to foster, and thus are less able than we would like, to assess their students' progress and to intervene effectively in their learning processes.
-- our own belief that teachers must and can participate in determining the educational values of using Logo in their classrooms and share in shaping its assesment and evaluating its place in education

For the purpose of our research we have decided to focus on areas that we call "critical aspects of Logo learning," that is, aspects of Logo which are essential if students are to reap the full benefit of their Logo learning experiences, and which most students usually do not

"discover" on their own, without timely interventions by a knowledgeable teacher.

As an example of the types of learnings we are focussing on, let's examine a Logo program written by a sixth-grade student in a Boston suburb. The teacher, who teaches Logo as part of his math curriculum, showed his students a series of geometric figures involving circles and arcs, and challenged them to reproduce one or more of them using the Logo turtle. The challenge shown at the top right of Figure 1, led to the design at the bottom right.

```
TO A
FULLSCREEN
HT
LT 90
REPEAT 180 [FD 57.5  BK 57.5  RT 1]
PC 0
REPEAT 180 [FD 57.5  BK 57.5  RT 1]
PU
HOME
LT 90
FD 28.75
PD
REPEAT 180 [FD 28.75  BK 28.75 RT 1]
PU
FD 57.5
PD
PC 1
REPEAT 180 [FD 28.75  BK 28.75  RT 1]
ST
PU
FD 57.5
FD 28.75
RT 90
PD
REPEAT 360 [FD 1 RT 1]
END
```



Teacher's Challenge

Jimmy's Design

Figure 1

Looking at Jimmy's work with the teachers collaborating in our research seminar, we considered the following questions:

-- What important aspects of Logo are demonstrated in this procedure?
-- What would we like to ask the student about it?
-- What would we like to ask the teacher about it?
-- What next steps do we suggest for this student?

Although his teacher described Jimmy as an exceptional sixth grade math student, an

examination of Jimmy's procedure, showed that there were some important things that Jimmy did not know about using Logo effectively. For one thing, he did not give his procedure a meaningful name that would help him to remember or another reader to understand, what the procedure was supposed to accomplish. For another, he used the statement REPEAT 180 [FD *some distance* BK *the same distance* RT 1], four times, without using variables, or naming it as a subprocedure. And then there was his use of the HOME command. This made it easy for Jimmy to move the turtle back to the place it had started, but at the same time, made it impossible to place the design anywhere else on the screen, or to draw it in a different position.

On the other hand, there are some important mathematical learnings that Jimmy's procedure does demonstrate, including the idea that a "total turtle trip" is 360 degrees, and that a turtle rotation of 180 degrees can be used to fill a semi-circle. Furthermore, Jimmy must have done some careful experimenting to determine that the radius of a circle made by REPEAT 360 [FD 1 RT 1] is 57.5 turtle steps, and applied some mathematical reasoning to determine that the radii of the smaller semi-circles are half that value.

Next steps for Jimmy might include helping him learn to use some *critical aspects* of Logo such as *subprocedures* with *variables*, and to *name* his procedures effectively. Perhaps the most important thing for Jimmy to understand is how to *write procedures that other people can read* easily so they can enjoy the elegant designs he creates and use his ideas as models -or modules- for their own projects. The small change of rewriting procedures to be readable by and understandable to another person - or community - may be one that produces exponential results. *Naming, subprocedures, variables* and *programming style* form part of our preliminary list of *critical aspects of Logo learning*.

Charting the Logo Process

We are also interested in helping teachers understand and observe the processes students engage in as they solve Logo problems. To this end, we are beginning to identify the stages of a "Logo Process", similar to the stages of the "Writing Process" articulated by Donald Graves and others. In a recent collaborative problem-solving activity engaged in by teachers in our seminar, we observed the following stages:

1. Brainstorming
2. Choosing a project
3. Simplifying the project and making a plan
4. Identifying tasks
5. Creating first drafts of the different parts
6. Putting it all together
7. Embellishing and extending
8. Polishing and refining the programming
9. Publishing and sharing

Now we and our collaborating teachers are using this model to observe the work of their students. We hope to articulate and elaborate this model until it is as helpful to students and teachers of Logo, as the Writing Process has been to students and teachers of writing.

Step-by-Step

No one person or group can make Logo a success. As students, teachers, teacher educators, school administrators and researchers, all of us are doing our part to reach our ultimate goal: the improvement of teaching and learning, preparing the young people of our society to grow up into the twenty-first century. We know that by using Logo students can learn to be mathematitians, problem solvers, thinkers and effective communicators of their ideas. They can learn about learning and learn to be eager life-long learners. But will they? It's up to all of us in the wider Logo community. Like turtles, we make progress one step at a time. Working together and sharing our individual steps, we can achieve some of the promises made for Logo. We are reminded of a song we learned in the 1960s, which was sung as a round during the Civil Rights movement:

Step-by-step the longest march,
Can be done, can be done,
Many stones can form an arch,
Singly none, singly none,
And together what we will,
Can be accomplished still,
Drops of water turn a mill,
Singly none, singly none.

Symbolic Mathematics and LOGO

James E. White, Ph. D.

Bluejay Lispware

Abstract:

This paper discusses some ways in which symbolic mathematics and LOGO can mutually reinforce each other. The setting for this discussion is a computer language called CAL: The Mathematics Discovery Language.

CAL shares with LOGO the aim of inviting students to be active and cognizant participants in the construction of their knowledge. With it, teachers or students can create mathematical microworlds (called ENVIRONMENTS) in which students can explore new mathematical ideas, or test old ones, in virtually limitless ways. One of the ways in which CAL achieves this aim is by representing basic mathematical objects (such as curves, functions, or coordinate transformations) in such a way that students can examine them from both an Algebraic and a Geometric point of view. CAL contains LOGO. But its ability to represent mathematical objects in a mathematically NATURAL way makes it possible to explore interactions between Algebra and Geometry (using LOGO concepts and techniques) in novel and surprising ways.

This talk will illustrate with the following examples.

- CAL has a Spherical Geometry ENVIRONMENT called GEOM. In this environment, the turtle lives on a sphere. It moves in "straight" lines (or geodesics) on the sphere in response to the turtle commands, and thus illustrates some of the powerful facts of Riemannian Geometry discussed for example in Abelson & DiSessa's book Turtle Geometry [1]. In the GEOM environment, one may write "Spherical LOGO" programs that are full of surprises!

- Algebraic Exploration has been difficult to support with microcomputers until recently. With the second example, we step away from LOGO and show how CAL can be used to DERIVE and PROVE a formula for the sum:

  1^5 + 2^5 + 3^5 + ... + N^5 in terms of N.

  Along the way, we take up such topics as Mathematical Induction, Synthetic Division of Polynomials, and Simultaneous Systems of Linear Equations.

- Physics often provides rich food for the mathematical palate. We study a simple problem on ballistic trajectories. Given a fixed speed, with what angle with the ground should we hit

a ball to make it travel the maximum distance? This leads to the notion of "parametrized families of curves". And we conclude with an example of a LOGO-like ENVIRONMENT for the exploration of normal families, envelopes, and general pencils of lines.

*****

A] Geometry on a Sphere

In ordinary LOGO, the turtle knows three things. It knows its location in the Cartesian Plane, its heading (an angle) and the state of its pen (penup or pendown). In the Spherical Geometry environment, the sphere is represented on the screen as the projection of a "globe" onto the plane from a point directly above the intersection of the "equator" (latitude = 0) and the "prime meridian" (longitude = 0). These curves appear as perpendicular lines dividing the projection (a circle) into four equal quadrants. For orientation, the meridians and the latitude lines at +/- 45 degrees are also drawn.

The turtle in this environment represents its location as a pair of angles: latitude and longitude where these vary from -90 degrees to 90 degrees. Its "heading" is initially due East and its speed is 1. For convenience, the radius of the sphere is taken to be 10. The turtle's pen may be up or down.

Now there is NO absolute notion of "angle" which can be globally defined continuously on a sphere. This follows from a deep fact of topology called the Lefschetz Fixed Point Theorem. One might imagine the attempt to make such an assignment, say equipping the turtle with a compass which always points to the angle 0. But such an attempt must fail, as the turtle would discover if he visited the North Pole! So our turtle must use a relative notion: As he travels, he always knows which way is FORWARD, except at the poles he can ROTATE (in the counterclockwise sense, viewed from above) through any angle from that heading. Ignoring the singularity at the poles, the turtle also has a pair of numbers that otherwise unambiguously describe its heading. These are the Latvelocity and Longvelocity, the rate at which it is crossing the latitude and longitude lines. The turtle can also multiply its speed by any positive or negative factor. Finally, the turtle can "hop" to a new point on the sphere keeping its latvelocity and longvelocity.

Of course, the straight lines on the sphere are the arcs of great circles, and they have the property that, locally, they are the shortest arcs connecting pairs of points. Physically, these are the paths that a point mass would follow if it were given a push and forced to move in the sphere subject to no other forces.

Many beautiful experiments are possible with this spherical turtle. For example, in LOGO programs, the turtle can draw various "polygons" on the sphere. The sum of the angles of a geodesic triangle (measured in radians) is always larger than pi. The excess is the "area" the triangle encloses on a unit sphere. In the illustration we demonstrate this for a triangle whose side length is 5*pi. The program is written in spherical LOGO:

```
to demo () (repeat 3 (sforward 15.70)
                     (srotate 90)) ]
```

CAL's spherical LOGO makes more use of numeric than symbolic mathematics. It solves a differential equation (the geodesic equation) to implement the "sforward" or Sphere-forward and, in this case, there is little need for symbolic operations.

On the other hand, CAL can solve ANY second-order differential equation in the plane. The geodesic equations of motion are

just a special case of these. Often they correspond to classical systems with conservative "force laws". In the general case, it is possible to experiment with such systems in a manner similar to that sketched above for spherical geometry. In order to handle such equations CAL differentiates them symbolically and handles such things as the variation of parameters in a similar way. The accompanying illustration shows how a marble would roll in a quadratic dish under the force of gravity, given the right starting conditions. Another useful exercise would be to show Keplerian (planetary) orbits in an inverse square force law.

B] Algebraic Exploration

One of the most endearing features of LOGO is the fact that it encourages experimentation and exploration. Play is one of the most powerful and permanent sources of new knowledge. CAL imitates this pedagogical strategy in a new arena: Symbolic Algebra. The following is the scenario of just such a session.

Consider the sum of the first N positive integers. One finds in most calculus texts the formula:

```
1 + 2 + 3 + ... + N  =  N^2/2 + N/2
```

This expresses the sum of the first N positive integers as a Quadratic (degree 2) polynomial in the variable N. One also often sees the formula for the sum of the first N SQUARES:

```
1^2 + 2^2 + 3^2 + ... + N^2  =  N*(N+1)*(2*N+1)/6
```

This expresses the sum of the first N squares as a Cubic (degree 3) polynomial in N. An interesting conjecture is the following:

Is the sum of the first N kth powers expressible as a polynomial function of N of degree K+1? If it is, what is the factorization of the polynomial?

When one of my students tackled this question (for k=5) this was the result. We reasoned that if there was a formula for k=5 it must have this form:

```
1^5 + 2^5 + 3^5 + ... + N^5  =
            a*N + b*N^2 + c*N^3 + d*N^4 + e*N^5 + f*N^6
```

with the (undetermined) coefficients: a,b,c,d,e,f

Thus we defined a (symbolic) function

```
sum(N) = a*N + b*N^2 + c*N^3 + d*N^4 + e*N^5 + f*N^6
```

and gave CAL the system of 6 linear equations in a,..,f

```
sum(1) = 1^5
sum(2) = 1^5 + 2^5
sum(3) = 1^5 + 2^5 + 3^5
sum(4) = 1^5 + 2^5 + 3^5 + 4^5
sum(5) = 1^5 + 2^5 + 3^5 + 4^5 + 5^5
sum(6) = 1^5 + 2^5 + 3^5 + 4^5 + 5^5 + 6^5
```

in just that form. CAL solved the system and reported

```
f = 1/6
e = 1/2
d = 5/12
c = 0
b = -1/12
a = 0
```

The answers were given as rational numbers (most of CAL's arithmetic is EXACT and RATIONAL although it supplies up to 255 significant places in decimal calculations if asked).

We then told CAL to substitute those values for the previously undetermined coefficients. Thus we were able to evaluate the function sum NUMERICALLY, and checked that in fact for sum(N) = N^6/6 + N^5/2 + 5*N^4/12 - N^2/12

```
sum(7) = 1^5 + 2^5 + 3^5 + 4^5 + 5^5 + 6^5 + 7^5
```

At this point, the student was pretty well convinced that it was the correct formula for all N, but we decided to check it using Mathematical Induction. Doing the symbolic calculation,

CAL showed:

sum(N) + (N+1)^5 = sum(N+1)

Finally, when CAL graphed the function sum(x) we were able to see that it had a tangency at x = -1, so that likely (x+1)^2 divided sum(x) = x^6/6 + x^5/2 + 5*x^4/12 - x^2/12

We asked CAL to do the synthetic division and it gave the quotient: x^2/6 + x/6 - 1/12 with remainder 0 !
When we gave the quadratic to CAL's Expert system, it found the exact solutions x = +/- sqrt(3/4) - 1/2
And so we had our factorized formula:

1^5 + + N^5 = N^2 * (N+1)^2 * (2*N^2 + 2*N - 1)/12

In this exploration, we had available an algebraic "microworld" (Papert [2]) that the student could develop incrementally and explore

C] Parametrized Families of Curves

Returning to LOGO-like geometric exploration, if we ask what angle we should hit a baseball at so that (with a fixed speed) it travels the greatest distance, we see that there are several ways to answer it.

One way to do it with CAL is to define the curve:

height(time) = V*sin(theta)*time - 16*time^2

where V is the speed and theta is the angle the ball makes with the horizontal. Eventually, we would discover that the distance travelled is D = V^2*sin(2*theta)/32, and using a little calculus, see that it's maximized when theta = 45 degrees.

A more illuminating way is to define the CURVE for CAL:

trajectory(time) = (V*cos(theta)*t , V*sin(theta)*t - 16*t^2)

This is actually a parametrized family of curves (with parameters V and theta). This can be defined to CAL as a symbolic expression, and then values for V and theta can be supplied and changed interactively so that graphs can be superimposed and compared.

The illustration shows this for V=32 and theta = 30, 45, and 60 degrees.

CAL treats its mathematical objects in a natural way. They can be combined, passed to other objects and operated on, just as we expect students to do it. This makes it possible for them to test their ideas with CAL. For example, one can define arbitrary curves, and arbitrary coordinate transformations and then apply the transformations to the curves, or compose the transformations in

various ways. We discuss a few LOGO-like functions that can illustrate this point, and that require symbolic mathematics.

The ENVIRONMENT we describe here is useful for examining such notions as curvature of plane curves, focal points, and generic singularities of plane maps.

Start with a curve. To fix ideas, let it be the parabola

$$C(t) = (t , t^2)$$

At every point of this curve, we can define a "unit normal" vector, a vector emanating from the point (on a prespecified side) that is perpendicular to the curve. We thus have a "pencil of normal lines" determined by these vectors. Such a pencil of lines is a parametrized family of lines (parametrized by t). They capture interesting geometric properties of the curve, for example, the focal points correspond roughly to where light would focus if we imagined the curve to be a 1-dimensional mirror.

CAL has a function called NORMAL that, starting with a curve such as C and a variable d creates a family of curves, parametrized by d, where each curve is what is obtained by "pushing" C "d" units along its normal line. It has another function called FILL which, given two curves, draws any number of lines requested connecting corresponding points. Thus, the family of normal lines of any desired length could be drawn by constructing the new curve with NORMAL and then using FILL to draw the lines. Another interesting thing to do is to write a short LOGO-type program that draws the successive stages obtained by "pushing" the curve farther and farther along its normal lines. This is illustrated below. The singularities that appear in the pushed curves trace out a new curve for which the normal pencil is the envelope.

CAL has another function in this group called TANGENT which is like NORMAL, except that it constructs unit TANGENT vectors instead of NORMAL vectors to the original curve. These functions form another "microworld" in which specific geometric ideas may be explored.

References:

1. Abelson, H. and DiSessa, A. Turtle Geometry
MIT Press, Cambridge, MA, 1981

2. Papert, S. Mindstorms: Children, Computers, and Powerful Ideas
Basic Books, New York, 1980

Kids to Kids: USA and USSR

Cheryl A. Wissick, Instructional Technology Specialist
Albemarle County Schools
Charlottesville, Virginia

Kids To Kids was a project that came out of Danaan Parry's work of six years with the Earthstewards Network. Parry is a conflict resolution specialist who has worked in Northern Ireland, the Middle East and the Soviet Union. An aspect of his work involves citizen diplomacy; arranging people-to-people visits. After much planning and negotiating, Parry was able to schedule an unprecedented trip for twenty Soviet teenagers and their teachers. While visiting three cities, Washington, San Francisco and Seattle, the group lived with American families and was introduced to computer education. The Soviets were eager for this trip to become a reality as computer use in their country is not widespread. Although a computer literacy course is taught in the ninth and tenth grades, few computers are actually available to students. The adults and the students who visited were able to see how computers were used in the United States as they worked with other teens to produce computer music, develop graphics, and telecommunicate.

After the group returned home, a delegation of American teen-agers and adults visited them in their schools and homes. They were guests of Academician Andrei Ershov of the Soviet Academy of Sciences in Academgorodok (the Academic City) near Novosibirsk, Siberia. As a member of the Earthstewards Network, I was involved in the Washington arrangements and priviledged to be part of the tour to the Soviet Union. This is an account of a journey with kids, computers, and creative peacemaking!

May 1986: I attended my first meeting with the Washington group of the Earthstewards Network. We began initial plans for all the computer activities, tours, housing and transportation.

June 1986: Danaan had just returned from the Soviet Union. He extended Dr. Ershov's invitation to visit the Computing Center in the Academic City in Siberia. At that time I was a particicpant in the UVA-IBM Institute working with Logo and other learned based tools so I was excited when Danaan mentioned Professor Ershov's interest in Logo.

September 1986: We received final confirmation of the dates for both exchange trips. With only a month to go detailed planning intensified.

October 28, 1986: The Soviets arrived in Washington and spent the day with their host families.

October 30: After a day of touring and working off jet lag, the delegation was introduced to computer instruction. During the morning the students worked with RS-100 computers, which had been donated for use during their visit. As the moderator for the adult program, I had invited Dr. Glen Bull and Dr. Paula Cochran from the University of Virginia to join us. The adults listened to Dr. Lawrence Fedewa discuss aspects of computer education in the United States. The Soviets were concerned about the way computers were being used and if they were integrated into the curriculum.

Although the facts were interesting, we wanted to work with the kids. Just before lunch I had a chance to demonstrate IBM's Listen to Learn. The kids were hesitant at first but soon realized that they could type many words in different languages for the computer to speak. During lunch Glen and Paula discussed the aspects of Logo with some of the kids. They then managed to "kidnap" a few to create some Russian poetry using Logo poetry tools.

When they returned to our room we were ready for a demonstration break. The adults soon entered into the creative process as we added additional Russian words along with synthesized speech to the poetry. The resulting poems were a mix of Russian and English thoughts on a holiday common to both cultures.

October 31: The students visited George Mason University to work with the computers in four different departments. They were introduced to high speed graphics and demonstrations of on-line searching for library materials.

November 2: At George Washington University Andrei Ershov and a few of his students met with Dr. Dianne Martin and Dr. Rachelle Heller to discuss computer science projects.

November 4: The Soviet delegation left Washington to visit San Francisco and then Seattle.

November 16: Tired travelers returned to the Soviet Union!

November 21-23: Twenty-six American travelers, adults and teen-agers with interests in computers, departed for the Soviet Union. We experienced Moscow with midnight walks on Red Square, bus trips around the city, tours inside the Kremlin, subway excursions to Gorky Park and seating at the Bolshoi Ballet. While entertaining Russian friends at dinner, we discussed Logo along with a number of other projects involving computers, children and future exchanges.

November 24: We were invited to the Moscow Computer Club. In Moscow students have little access to computers, high school classes even have to travel to Moscow University to use the computers. The club was founded to bring computers to the students who were learning about computer programming without the benefit of a machine. In a short time after it opened over 500 students asked to join. Since the club was in its new facilities only 3 computers were available for the students to demonstrate their projects and for us to exchange notes on Logo programming.

November 25: Off we flew to Novosibirsk through four time zones in four hours, spending most of our day traveling. We were greeted by our Russian friends and escorted to the Academic City's hotel, which is normally reserved for visiting scientists. The air was crisp and snow was falling as we had our first taste of Siberia.

November 26: After a morning sightseeing and meeting the mayor of the district, we were able to spend the afternoon in the Academic Computing Center. A greeting and presentation by Academician Ershov preceded our tour of the center. We were interested in the different keyboards and the variety of computers in use. Even though we were introduced to current projects, we had little time to discuss details as the tour continued at a quick pace.

We had two more events on the schedule for that day. We stopped at the Young Technicians Club to see projects in electronics, model designing, shipbuilding, and mechanics. Dinner that night was arranged for socializing with individual families. We were all escorted to different homes and treated to wonderful native Siberian dishes.

November 27: Several school visits were arranged for the day. We met with the headmaster, students in an English class, and members of the Friendship Club. After exchanging photos, letters and art work, we were given a tour of the school. Although the emphasis in this school was on English, with students beginning instruction in second grade, a computer lab was also in use where students were working on Basic programming and English grammar software.

The day went by quickly as we walked through the woods to our various meetings with the Fencing Club, Young Scientists Club, University president, Math & Physics High School principal, and students of the University. Everywhere we went we were followed by the camera men and reporters! The American youth in our group were the first ever to visit the city and be exposed to this world of computer technology and education.

We were only allowed to spend two days in Novosobirsk so the families prepared a special banquet for us on our Thanksgiving Day. After feasting on special pastries and cavier, we shared singing, dancing, and poetry writing. The feeling of belonging and new found friends only made it harder to leave, not knowing when we would be able to return.

November 28: The flight back to Moscow gave us four more hours to enjoy in our day. We were able to visit the Pioneer Palace to see the various clubs along with a presentation of the Theatre Club. Our teen-agers played charades, danced, and sang along with the Soviet teens to create a joint video/film.

November 29: This was the end of journey and I left with more questions about computers than answers. At this point we were all so tired we couldn't even talk straight and it was even worse after the twenty-four hour trip back to Washington. Although the exchange was only the beginning for future trips and more work with Logo, we had been able to share Logo curriculum guides, materials and magazines. We hope to continue the work using the postal service and Logo articles, looking to the day when trips will be longer, more computer intensive and possibly even take place on electronic mail!

TURTLE STEPS TO SINGLE SWITCH COMPUTER ACCESS

Cheryl Wissick, Albemarle County Schools, Virgina
Betsy Calvert, Fairfax County Schools, Virginia

Introduction:

From June 16 through July 5, 1986, twenty special educators worked together at the UVA - IBM Institute. The Institute became a vehicle for the meeting of minds from across the country encompassing all the areas of special education. As participants in the Institute we experienced Logo and other learner based tools during the three week project, which combined the expertise of the University of Virginia's Dr. Glen Bull and Dr. Paula Cochran with IBM's David Cartmell. Learner based tools allow the educator to modify programs as appropriate for the students, thus providing the students more control over their environment. Special interest groups focusing on particular handicapping conditions developed. As educators interested in computer access for severely impaired and multihandicapped students, we created several single switch Logo procedures. We also gave thought to the justification and the hierarchy of skills necessary for these students to have successful computer access.

Justification:

For students who are cognitively impaired and multihandicapped, a great deal of instructional time is spent in establishing cause and effect relationships. Fans, radios, bells and other environmental objects can be useful for training these students to understand the result of pressing a switch to control an object. For these students the primary goal is to have some direct control over their environment. Tapes, radios, musical toys, bells, and buzzers offer a wide range of switch elicited auditory stimuli. These devices typically include an auditory component directly controlled by students. A survey of equipment available for the severely handicapped suggests a need for visually stimulating activities which can also be directly controlled by the students. The only visual stimuli previously available for students were television, movies, photographs or slides but these did not offer direct interaction without a great deal of modification by the teacher.

The computer serves as a valuable tool to use with multihandicapped individuals. User access can be controlled by a single switch, input can be modified by the teacher and student effort can be immediately reinforced through dual modalities. The development of learner based tools makes it possible to modify procedures easily. Also, Logo graphics contributes to user attention and interaction.

A pilot study, conducted during the 1985-86 school year in one of our classrooms in Fairfax County, Virginia examined the visual attending skills of multihandicapped students when presented with stimuli produced by a television and by a Logo graphics program. Students were shown a 4 minute 30 second Logo turtle graphics program written by the teacher and an equal time segment of a television game show. Segments were presented one per day for four days each for a total of eight days. A comparison was made between the length of time during the presentation that the students focused visually on the two types of stimuli and the percentage of visual focusing observed every 10 seconds during the time period .

The eighteen students in the study ranged from ages five to twenty one. They were non-verbal and classified as severely impaired and multihandicapped. Data was collected during school hours in the classroom setting. Although the students were positioned behind a screen there was still a high level of auditory distraction and many classroom interuptions. Care was taken to reduce other environmental and personal distractors, such as hunger, fatigue and wetness. The data was collected by the teacher and the aide who positioned themselves behind the screen and did not prompt the students.

Two types of data were collected: regular interval measurement using 10 second intervals (27 possible intervals) and measurement of the amount of time on-task measured in seconds (270 possible seconds). Results were collected and averaged for each student for both types of data and both programs. The results for the whole group were divided according to age- under and over 12 years.

Summary of results:

Total Group
Average On-Task Performance

	Interval		Percentage	
	Mean	SD	Mean	SD
Logo	63.8	27.5	62.8	25.9
TV	62.6	29.2	55.6	28.3

Although the results do not show significant differences, they do demonstrate that Logo graphics compared to commonly used television programming can be as equally stimulating to these students. Along with teacher observation, these results supported the development of Logo procedures to use with severely handicapped students for the purpose of reinforcing visual attending and for teaching cause and effect.

Skill Sequence and Computer Programs:

The next task was to establish a hierarchy of skills for computer access and to provide the teacher with sequential objectives for teaching single switch computer use to cognitively impaired students. The following hierarchy and Logo procedures were developed:

1) The student will visually attend to the screen for a short program.
   The procedures developed for this objective generate a picture on the screen. The turtle stops at various points and plays a tune.

2) The student will press a switch and receive visual or auditory feedback as long as the switch is depressed.
   The student presses the switch and random shapes, colors and tunes are displayed. Once the student releases the switch the screen becomes blank. The teacher could then prompt the student to press the switch again. The objective is to develop motivating action on the screen to encourage the student to keep the switch pressed.

3) The student will press a switch and receive visual and auditory feedback with every switch press. Using verbal prompting the teacher will train the student to press the switch after each turn.

Once the student is able to press the switch, the teacher can train the student to press the switch for every change. Many programs require students to press and release the switch to make a choice. With these procedures the student receives a color change and an auditory reinforcement with every switch press. If the switch is held down it will continue to change. When the switch is released the picture remains on the screen but additional changes do not occur until the switch is pressed again.

4) The student will receive a verbal prompt from the computer to press a switch and to begin a brief program. After the screen clears the student will be prompted to press the switch to begin again.

The turtle generates random designs on the screen. After the design finishes, the computer gives the student verbal reinfocement and tells the student to go again. Two programs were developed, in one the screen goes blank after each design and in the other one the picture continues to build.

5) The student will be prompted by the computer to press the switch to move the turtle through a maze.

A maze is drawn on the screen. The computer will randomly select one of the two paths to travel. At each turn the computer asks the student to press the switch to have the turtle continue through the maze.

6) The student will be shown individual pictures to represent familiar items or actions. The student will press the switch to indicate his choice. The computer will then signal the teacher to assist or reward the student.

The program presents each picture on the screen and gives a verbal question. (ie: "Do you want something to drink?") When the student presses the switch the computer plays a tune to signal the teacher and then responds to the question (ie. "Yes I want something to drink.") This could also be the beginning of a simple communication program.

7) The student will be given verbal commands and will press the switch to indicate selection of the verbal command. The command will then be executed on the screen.

This program allows a single switch user to experiment with Logo graphics. Logo commands are presented verbally one at a time. The turtle executes the command, when the student presses the switch.

8) The student will press a switch to indicate selection of two scanned items.

This program was designed as a teaching tool to introduce scanning arrays. The teacher chooses from either 2, 3, 6, or 8 scanning boxes and attaches an overlay or pictures to the screen. The student is then instructed to select an item. The student gains practice in pressing the switch to make the correct selection.

Summary:

These objectives and procedures were just the beginning of the possibilities available for us as special educators working with Logo. We continued our work with other members of the Institute through electronic mail, modifying the procedures for individual students and creating additional programs for each objective. The computer offered us many new possibilities with which to work with our students who are cognitively impaired and multihandicapped. This range of possibilities has increased due to the use a of learner based tool such as Logo.

Speakers

Eadie Adamson
The Allen-Stevenson School
132 East 78th Street
New York, NY 11021

Tom Addicks
Berkshire Country Day School
Box 867
Lenox, MA 01240

Steve Adelberg
Bailey's Elementary School
6111 Knollwood Drive
Falls Church, VA 22041

Scott Anderson
Broderbund Software
17 Paul Drive
San Rafael, CA 94903-2101

Christine Appert
Education Department
Children's Rehabilitation Center
2270 Ivy Road
Charlottesville, VA 22901

Suzanne L. Bailey
Fairhill Elementary School
3001 Chichester Lane
Fairfax, VA 22031

Hunter Ballew
UNC - Chapel Hill
Peabody Hall 037 A
Chapel Hill, NC 27514

Trent Batson
ENFI Project
Gallaudet University
800 Florida Ave NE
Washington, DC 20002

Jane Bauserman
Norfolk Public Schools
PO Box 1357
Norfolk, VA 23501

Alison Birch
Terrapin Inc.
222 Third Street
Cambridge, MA 02142

Louisa Birch
Meadowbrook School
Farm Road
Weston, MA 02193

Glen Bull
Curry School of Education
University of Virginia
Charlottesville, VA 22903

Betsy Calvert
11486 Links Drive
Reston, VA 22070

Jon Campbell
Whispering Oaks School
RR # 1
Mackinaw, IL 61755

Richard Carter
Lesley College
29 Everett Street
Cambridge, MA 02238

David Cartmell
IBM
43H-895 Neighborhood Road
Kingston, NY 12401

Ihor Charischak
445 Broadway, Apt 2K
Hastings-on-Hudson, NY 10706

David Chesebrough
Sewickley Academy
Sewickley, PA 15143

Paula Cochran
Curry School of Education
University of Virginia
Charlottesville, VA 22903

George Collison
Holyoke Public Schols
500 Beech Street
Holyoke, MA 01040

Sheila Cory
Chapel Hill - Carrboro City Schools
Lincoln Center
Chapel Hill, NC 27514

John Crowley
LEGO Systems Inc.
555 Taylor Road
Enfield, CT 06082

Bev and Lee Cunningham
PO Box L
Treynor, IA 51575

Lis D'Introno
Holyoke Public Schols
500 Beech Street
Holyoke, MA 01040

Andrew Evans
5141 Hagerstown Hall
University of Maryland
College Park, MD 20742

Mary Fairless
Tennessee State University
10th and Charlotte
Nashville, TN 37203

Leslie Farlow
Curry School of Education
University of Virginia
Charlottesville, VA 22903

Dorothy Fitch
Terrapin Inc.
222 Third Street
Cambridge, MA 02142

Doris Flowers
Curry School of Education
University of Virginia
Charlottesville, VA 22903

America Foster
George Washington Carver High School
PO Box 837
Fieldale, VA 24089

Jacqueline Foster
Route 5, Box 675
Martinsville, VA 24112

Pam Freyd
University of Pennsylvania
Graduate School of Education
Philadelphia, PA 19104

James Fry
Novi Woods Community School
25195 Taft Road
Novi, MI 48050

Michael Gelman
19 Darlington Drive
Wayne, NJ 0740-2805

Bill Glass
Harvard Associates Inc.
260 Beacon Street
Somerville, MA 02143

Jonell Glass
205 B West Washington Street
Blacksburg, VA 24060

E. Paul Goldenberg
Education Development Center Inc.
55 Chapel Street
Newton, MA 02160

Stephen Hain
Coral Software
PO Box 307
Cambridge, MA 02142

John Hall
Eastconn Vocational Center
Columbia, CT 06237

Janet Hanson
College of Education
University of South Florida
HMS 419
Tampa, FL 33620

Judi Harris
Curry School of Education
Department of Educational Studies
University of Virginia
Charlottesville, VA 22903

Jane Harrop
Curry School of Education
University of Virginia
Charlottesville, VA 22903

Francis A. Harvey
Lehigh University
College of Education
301 Broadway, 3rd Floor
Bethlehem, PA 18015-1598

Catherine Helgoe
LEGO Systems Inc.
555 Taylor Road
Enfield, CT 06082

Rachelle Heller
George Washington University
Dept. of Electrical Engineering
Washington, DC 20052

Gretchen Kelly
Curry School of Education
University of Virginia
Charlottesville, VA 22903

Philippe Krakowsky
Coral Software
PO Box 307
Cambridge, MA 02142

Rosanne Kramer
Brøderbund Software
17 Paul Drive
San Rafael, CA 94903-2101

Patricia Lang
Fairview Elementary School
Cheyenne, WY 82001

Dianne Lawrence
Consortium for Interactive Instruction
WHRO Public Broadcasting
5200 Hampton Blvd.
Norfolk, VA 23501

Gayle Lawrence
Brittonkill Central School
Melrose, NY 12121

Jean Lindsay
Curry School of Education
University of Virginia
Charlottesville, VA 22903

Segano Litheku
Curry School of Education
University of Virginia
Charlottesville, VA 22903

Tom Lough
Curry School of Education
Dept of Educational Studies
University of Virginia
Charlottesville, VA 22903

C. Dianne Martin
George Washington University
Dept. of Electrical Engineering
Washington, DC 20052

Laurie McCullough
Curry School of Education
Dept of Curriculum and Instruction
University of Virginia
Charlottesville, VA 22903

Kevin McPeak
Whispering Oaks School
RR # 1
Mackinaw, IL 61755

Sean McPeak
Whispering Oaks School
RR # 1
Mackinaw, IL 61755

Hal Melnick
Bank Street Graduate School of Education
610 West 112th Street
New York, NY 10025

Janet Moore-Coll
Curry School of Education
Dept of Educational Studies
University of Virginia
Charlottesville, VA 22903

Stephen Mosher
Holyoke Public Schols
500 Beech Street
Holyoke, MA 01040

Robs Muir
Claremont Unified School District
Claremont, CA 91711

Dan Novak
Dept of Math and Computer Science
Ithaca College
Ithaca, NY 14850

Barbara Nye
Tennessee State University
10th and Charlotte
Nashville, TN 37203

Judith Olson
Western Illinois University
Department of Mathematics
Macomb, IL 61455

Seymour Papert
MIT Epistemology and Learning Group
E15-309, 20 Ames Street
Cambridge, MA 02139

Rebecca Poplin
Barwise Junior High School
Wichita Falls, TX 76308

Peter Rawitsch
Glenmont Elementary School
Route 9W
Glenmont, NY 12077

Melda Richards
Curry School of Education
University of Virginia
Charlottesville, VA 22903

Curtis Ring
Wooster High School
Wooster, OH 44691

Teresa Rosegrant
Dept of Special Education
George Washington University
Washington, DC 20052

Andre Rossi
Harvard Associates Inc.
260 Beacon Street
Somerville, MA 02143

Brian Silverman
Logo Computer Systems Inc.
9960 Cote de Liesse Road
Lachine, Quebec H8T 1A1
Canada

Stephen Sesko
Science Education Center
Lawrence Livermore National Lab
PO Box 808, L-793
Livermore, CA 94550

Jerry Short
Curry School of Education
University of Virginia
Charlottesville, VA 22903

Gary Stager
12 Locust Place
Wayne, NJ 07470

Michael Tempel
Logo Computer Systems Inc.
555 West 57th Street, Suite 1236
New York, NY 10019

Chris Templar
Johnson Bible College
Knoxville, TN 37998

Leslie Thyberg
Learning Research and Development Ctr
University of Pittsburgh
3939 O'Hara Street, Room 314
Pittsburgh, PA 15260

Robert Tinker
Technical Education Research Center
1696 Massachusetts Avenue
Cambridge, MA 02138

Steve Tipps
Midwestern State University
Wichita Falls, TX 76308

Joyce Tobias
38 Maple Avenue
Lexington, MA 02173

Dr. Rena Upitis
Faculty of Education
Queens University
Kingston, Ontario K7L3N6
Canada

Christopher Van Leeuwen
Holyoke Public Schols
500 Beech Street
Holyoke, MA 01040

Madge Vining
East Connecticut State University
Willimantic, CT 06226

Dan and Molly Watt
Educational Alternatives
Gregg Lake Road
Antrim, NH

Barbara West
Tennessee State University
10th and Charlotte
Nashville, TN 37203

James White
Bluejay Lispware
814 East Woodland Lane
Glendora, CA 91740

Cheryl Wissick
Albemarle County Schools
Charlottesville, VA 22901

Susan Wolff
1608 Stowe Road
Reston, VA 22094

Index to Papers by Session

In the following index, papers are listed by author within each session. General sessions are listed first, followed by regular sessions numbered one through twenty-one, then paper sessions numbered one through nine.

General Session: ECLC '87 Kickoff

Watt, Daniel H., and Molly Watt, 207

Logo Gala Social

Rawitsch, Peter
BUDGET-MINDEDSTORMS, 166

Regular Sessions

Session One: Logo and Music

Upitis, Rena B.
Logo-Like Music Environments, 195

Fry, James
Logo Music Tools: "MusicLand", 87

Session Two: Logo Interactions

Bauserman, Jane, and Dianne Lawrence
One Computer—Thirty Kids—And You Want Me to Use Logo??, 23

Muir, Robs John
Apropos Logo and Telecommunications, 141

Session Three: Logo: FD to the Future

White, James E.
Symbolic Mathematics and LOGO, 212

Krakowsky, Phillipe, and Stephen Hain
(No paper submitted)

Session Four: Logo Connections

Bull, Glen L., and Paula S. Cochran
A Book with 50,000 Pictures: Logo and Videodiscs, 34

Cartmell, David J.
Talking with IBM Logo, 43

Session Five: Logo Environments

Birch, Alison
Creating a Learning Environment for Teachers, 27

Thyberg, Leslie F.
The Basis for a Process Approach to Programming in Logo, 183

Session Six: Secondary Logo Integration

Foster, America H.
Creating a LOGO Environment in a Geometry Class, 75

Muir, Robs John
Secondary Logo Opportunities, 146

Session Seven: Logolike Software and Other Compelling Media

Kramer, Rosanne
Fantavision in Education (Paper by Scott Anderson), 11

Melnick, Harold R.
Tools for Making Meaning in Mathematics—A Rationale, 136

Session Eight: Special Goldenberg Session

Goldenberg, E. Paul
Making Functions Concrete: Explorations for Algebra and Pre-algebra, 104

Goldenberg, E. Paul
Language Science, Language Play: Using Logo to Explore What You Know, 99

Session Nine: Building with Logo

Addicks, Tom
Logo and Computer Assisted Design (CAD), 3

Fitch, Dorothy M.
Logo Learning Tools for the Classroom, 72

Session Ten: Logo and Robotics

Helgoe, Catherine
No Limits LEGO TC logo (Paper by John Crowley), 58

West, Barbara, Mary Fairless, and André Rossi
Let the Valiant Turtle Robot Lead the Way to Logo (Paper by Barbara Nye), 154

Session Eleven: Logo in the Early Years

Thyberg, Leslie F.
Young Users of Logo—The Challenge and the Opportunity, 187

Lawrence, Gayle
Logo: Where Models and Reality Connect, 123

Session Twelve: Logo International

Watt, Molly, and Daniel H. Watt
The Logo and Computers in Education Workshop: A Report on a Teacher Training Project at The People's Education Press, Beijing, China, 203

Martin, C. Dianne, and Rachelle S. Heller
Educational Computing in the Soviet Union, 128

Wissick, Cheryl A.
Kids to Kids: USA and USSR, 218

Session Thirteen: Special Session: Vertical Logo Integration

Chesebrough, David E.
Supporting a Successful Schoolwide Logo Integration into an Entire School System, 49

Session Fourteen: Elementary and Middle School Curriculum Applications for Logo

Harris, Judi
Logo LinX with Traditional K–8 Curricula, 115

Session Fifteen: Communicating through a Keyboard

Rosegrant, Teresa
Exploring the Algorithms of Written English, 171

Batson, Trent
The ENFI Project, 19

Session Sixteen: Thresholds and Ceilings

Adamson, Eadie
LogoWriter: A Learning Environment and a Programming Language for All Levels, 1

Muir, Robs John
No Threshold? No Ceiling?, 144

Silverman, Brian, and Michael Tempel
Logo Thoughts: LogoWriter and Logo II, 173

Session Seventeen: Student Presentations

Foster, Jacqueline M.
The Platonic Solids, 80

Gelman, Michael A.
Methods of Simulating a Robot's Behavior Using Logo, 91

McPeak, Sean, Kevin McPeak, and Jon Campbell
Student Designed Tutorials, 133

Ring, Curtis
Adventuring in Logo, 168

Session Eighteen: Microcomputer Based Laboratories

Tinker, Robert
Microcomputer Based Laboratory (No paper submitted)

Session Nineteen: From the Classroom

Cunningham, Beverly, and Lee Cunningham
Associated Learning: Logo Primitives in Real Life, 61

Birch, Alison
Playing with Language through Logo Projects, 31

Session Twenty: The Big Picture

Poplin, Rebecca L.
An Environment for Action, 163

Tipps, Steve
Logo Beyond the Square, 191

Session Twenty-One: Student Activities

(See paper by John Crowley for Session Ten)

Paper Sessions

Paper Session One

Cory, Sheila
The Mathematical Turtle: Logo in the 4–8 Math Curriculum, 54

Stager, Gary S.
If There Had Been Logo. . .I Might Have Learned Math, 175

Paper Session Two

Novak, Dan
Teaching Methods of Interfacing Logo Procedures, 150

Vining, Madge Alexander
SeeLogo: A New Perspective on an Old Friend, 199

Paper Session Three

Bailey, Suzanne L.
Logo Activities and Strategies in the Kindergarten, 13

Templar, Chris
Integrating Logo into the Elementary School Curriculum, 180

Paper Session Four

Wissick, Cheryl, and Betsy Calvert
Turtle Steps to Single Switch Computer Access, 222

Carter, Ricky
The Logo Algebra Project, 39

Paper Session Five

Freyd, Pamela
Reading, Writing, and Logo: The Ultimate Language Experience, 84

Ballew, Hunter
Geometry through LOGO for Teachers in the Middle Grades, 15

Olson, Judith, and Patricia Lang
Cross Over the Bridge—A Journey to Abstraction with Logo and Manipulatives, 158

Paper Session Six

Hanson, Janet Bard
The Turtle Grows Up: LOGO Applications in High School Physics, English, Biology, and Artificial Intelligence, 111

Goldenberg, E. Paul
Methods for Teaching Advanced Logo Topics (No paper submitted)

Silverman, Brian
Logo in the Microworld of Life, 172

Paper Session Seven

Cunningham, Beverly, and Lee Cunningham
Ten Tried and True Turtle Technical Tips, 65

D'Introno, Lis, Stephen D. Mosher, Christopher Van Leeuwen, and George Collison
A Logo Curriculum Based on Critical Thinking and Problem Solving Using the Polya Model, 68

Paper Session Eight

Moore-Coll, Janet, et al.
Evaluation of ECLC '87 (No paper submitted)

Adelberg, Steven M.
Creative Use of Logo with Special Student Populations: Using Atari Logo in the Primary Grades (at Bailey's Elementary School) with ESL Students, 8

Paper Session Nine

Harvey, Francis A.
Tool Procedures for Efficient Development of Logo-Based Applications
Text Screen Formatting Tools, 119

Glass, Jonell, and Andrew Evans
Logo Quickly, 96